UNIVERSITY OF SOUTH AUSTRALIA

School of Engineering

Development of an indoor localisation system based on ambient Wi-Fi and BLE fingerprinting and machine learning classification

Student: **Quang Loc Ngo**
ID Number: **110270717**
Supervisor: **Dr. Arek Dadej**

Engineering Minor Thesis for the Master of Engineering (Telecommunications)

November - 2020

ABSTRACT

Localisation system based on ambient Wi-Fi and BLE (Bluetooth Low Energy) is widely used in recent years. However, performance and localisation accuracy of deployment models are still key problems. In this project, the working demonstrator of localisation system based on ambient Wi-Fi and BLE (Bluetooth Low Energy) fingerprinting is developed to evaluate the system's performance and compare between methods. For the experimental deployment, the SparkFun ESP32 Wi-Fi and BLE supported boards are used as the terminals to receive Wi-Fi and Bluetooth signals, and simultaneously are as the Bluetooth signal transmitters. In the first step, the system collects ambient Wi-Fi and BLE fingerprinting data for designated reference points in an area of interest. Then, the data is used to train a simple machine learning model (using Matlab classifier and Neural Networks). The training is implemented separately for ambient Wi-Fi and BLE fingerprinting data, and for combined ambient Wi-Fi and BLE. The final step in the project involves evaluation of the system's performance. The experimental results demonstrates the good performance of indoor positioning system Wi-Fi based on Wi-Fi fingerprinting as well as the greater localisation accuracy in combined Wi-Fi and BLE model.

DECLARATION

I, Quang Loc Ngo, hereby declare that the minor thesis "Development of an indoor localisation system based on ambient Wi-Fi and BLE fingerprinting and machine learning classification" is my own, under the guidance of supervisor. This work carried out in the School of Engineering, University of South Australia, from February 2020 to November 2020. The data and results presented in the minor thesis are completely honest. The sources of references used in the thesis are mentioned in the references. This minor thesis contains 13,427 words, including tables, appendices and references but excluding diagrams and photographs.

Quang Loc Ngo

November 2020.

ACKNOWLEDGMENTS

Foremost, I profoundly thank my thesis supervisor, Dr. Arek Dadej, Program Director, Master of Engineering Telecommunications, University of South Australia, for his immense guidance and supportive encouragement. Without him, I cannot completed this thesis.

I am also grateful to the lecturers of the School of Engineering, University of South Australia for giving me valuable knowledge during my time here.

Finally, I would like to thank all loved ones in my family, friends and colleagues who have always helped and encouraged me to overcome the difficulties to complete this thesis.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

ACU	: Area Under The Curve
AOA	: Angle of Arrival
APs	**:** Access Points
BLE	: Bluetooth Low Energy
CE	: Cross Entropy
FPR	: False Positive Rate
GPS	: Global Positioning System
IDE	: Integrated Development Environment
IEEE	: Institute of Electrical and Electronics Engineers
ISM	: Industrial, Scientific and Medical
IPS	: Indoor Positioning System
KNN	: K-Nearest Neighbors
MSE	: Mean Squared Error
NB	: Naive Bayes
NLOS	: Non Line of Sight
NN	: Neural Net
RFID	: Radio Frequency Identification
ROC	: Receiver Operating Characteristic
RP	**:** Reference Point
RSS	: Received Signal Strength
RSSI	: Received Signal Strength Indicator
TDOA	: Time Difference of Arrival
TOA	: Time of Arrival
TOF	: Time of Flight

TPR : True Positive Rate

UWB : Ultra Wide Band

WLAN : Wireless Local Area Network

WPA : Wi-Fi Protected Access

WPS : Wi-Fi Protected Setup

CHAPTER 1: INTRODUCTION

1.1 Overview of indoor positioning system.

Location detection has become a popular field in recent years. With the development of technology, the global positioning system GPS has been pre-installed on most mobile devices; thus, and outdoor positioning and navigation become easier and more popular than ever. However, in indoor places, the GPS system faces difficulties such as weak signals and large noise, which reduces the accuracy of the results. Therefore, new methods and technologies are needed to meet this demand. Indoor positioning system (IPS) is a set of hardware and software solutions used to locate people or objects in indoor environments using radio signals, magnetic radiation, or sensor data from mobile devices [1].

To detect the position we can use numerous methods with different techniques and algorithms. The form of the acquired data can be signal strength or range, which is used to calculate the estimated position [2]. The proposed various techniques for indoor positioning could be divided into two types: radio-based positioning and non-radio-based positioning. The methods based on Radio Frequency Identification (RFID), Ultra Wide Band (UWB), Bluetooth, ZigBee, Frequency Modulation (FM), and Wireless Local Area Network (WLAN) are radio-based positioning category. Whereas non-radio-based positioning category apply infrared (IR), inertial systems, ultrasonic and sound techniques, visible light, and the geo-magnetic field [3].

There are various algorithms for location detection in indoor positioning, including triangulation, fingerprinting, signal propagation modeling, dead reckoning, proximity, cooperative positioning [2]. One of the most common and successful methods is fingerprinting. This technique determines the position by analyzing scene and compare it with the existing database. It consists of two phases: the first phase (offline phase) builds the matching database by surveying the area to collect location features. The database of location information is created by identifying the suitable unique signatures of the signals used. The second phase (online phase), the system compares the current signal factors with the pre created database to determine the location [4].

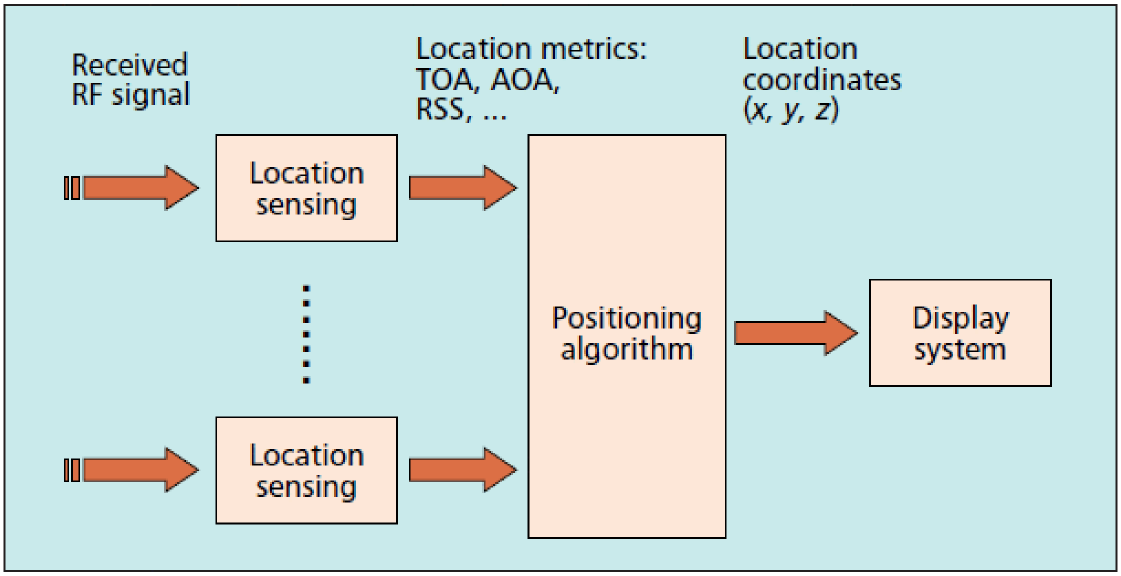


Figure 1: ***A block drawing of general indoor localisation system*** *[5]*

A block drawing of general indoor localisation system is described in above Figure 1. Location sensing, positioning algorithm, and display system are the main units of this system. The distance between a unknown object and an identified reference point (RP) is measured by location sensing devices via common methods of location estimation like time of arrival (TOA), angle of arrival (AOA) or received signal strength (RSS). These data are then processed using positioning algorithms (such as Fingerprint, Triangulation or Trilateration) to estimate the position coordinates of the objects. Finally, the users know the position of the object through the display system [5].

1.2 Applications of indoor positioning system.

The application areas of IPS are omnipresent in many fields of today's modern social life, namely logistics and automation, industry, environmental monitoring, home and consumption, emergency services and health care.

The applications at home are the systems that provide the support for daily activities of the elderly people, such as important signs monitoring, falling detection and detection of emergency. In environmental field, some phenomena such as air pollution, heat, pressure, humidity are observed by IPS applications. Indoor positioning capabilities also bring helpful assistance in rescue services and law enforcement. In many industries, location positioning is an essential element for automatic manufacturing, intelligent factories, monitoring automated and quality control. IPS applications also increase significantly efficiency on cargo management systems in airports, ports and railway stations [6]. In the hospital, an indoor positioning system is a promising method to meet the growing demands on medical services. In the hospital, there are the applications to monitor patients and equipment as well as the position of nurses, doctors in emergency situations [7].

1.3 Challenges of indoor positioning system.

Although indoor positioning has been an essential component in many applications, there is currently no standards for the IPS system. Instead various systems are tailored to meet the requirements of different applications, depending on required accuracy, spatial dimensions, building materials, and budget [8].

While the positioning technologies have evolved tremendously nowadays, there are still some typical challenges to overcome that are listed below [9]:

- Multipath effect and NLOS propagation
- Varying environment
- Full network coverage
- Real-time requirements
- Seamless switching between indoor and outdoor scenario.

In addition, for any information system, security is always a crucial issue. In particular, the network of location services based on measurement of signals outside the air is vulnerable to attacks. Thus, IPSs also face the common attack threats as False Node, Spoofing, Sinkhole, Replay, Wormholes or Denial of Service [10].

1.4 Description of the project.

The target of the project is to build the Wi-Fi and BLE fingerprinting based Indoor Positioning System. The operation is divided into two phase. In an offline phase, the data of reference points in the choose place is collected to create the Wi-Fi and BLE fingerprinting database. This data is used to train a simple machine learning model. In an online phase, the localization server based on the measured signal strength data of mobile object in the area of interest to estimate the its location, via classifier techniques. Finally, the project evaluates the system’s performance and compare between Wi-Fi and BLE methods. This thesis focuses on the demonstration in Wi-Fi fingerprinting based system.

CHAPTER 2: LITERATURE REVIEW

2.1 Wireless technologies.

Wireless technologies used throughout the world are mainly for communication and data access, not for location. However, researchers can still rely on properties derived from radio signals to estimate the location of the signal source and apply it to indoor positioning systems. Existing indoor positioning systems are often based on wireless technologies deployed in the operating environment.

2.1.1 Wi-Fi technology.

Wi-Fi is a popular wireless technology worldwide and also one of the most used technologies on the local area networking applications. It is deployed by installing access points (APs) that allow devices in the network to access wirelessly. This allows devices to move arbitrarily within the coverage of access points. This technology is developed base on wireless standard 802.11 of the Institute of Electrical and Electronics Engineers (IEEE). The Wi-Fi technology operates in 2.4 GHz, 5 GHz, 6 GHz and 60 GHz bands, each of which allows different operating ranges and performance. The latest generation of Wi-Fi connectivity is Wi-Fi 6, which is based on IEEE 802.11ax standard with higher data rates, increased capacity and better efficiency than previous technologies [11].

With Wi-Fi technology, signals are transferred in both directions from the access point or from the access device. This signal transmission feature allows to estimate the relative position between the device and the access point by using parameters such as signal strength, signal angle, propagation time of radio waves, thereby determining the location of the device in practice. Indoor positioning systems relying on Wi-Fi technology use access points to find out the target's location in a zone, the more access points the higher the accuracy. Fingerprinting and signal propagation modeling are the most common techniques for IPS implemented with Wi-Fi technology [2].

2.1.2 Bluetooth technology.

Bluetooth is a wireless technology standard used for exchanging data between mobile devices over short distance, operating in the 2.4 GHz ISM spectrum band. The 2.4 GHz frequency band enables a good balance between range and throughput, consumes very little power, in addition, it is available worldwide [12]. These features make Bluetooth a true standard for low-power wireless connectivity. It opens up the method of positioning in an indoor environment by installing some Bluetooth devices that act as beacons for the device to be located.

Bluetooth Low Energy (BLE) is a technology developed from Bluetooth, is called Bluetooth version 4.0, that is designed for very low power operation to save more energy for devices. The specifications of BLE are described in the table below:

Frequency Band	2.4GHz ISM Band (2.402 – 2.480 GHz Utilized)
Channels	40 channels with 2 MHz spacing
Channel Usage	Frequency-Hopping Spread Spectrum (FHSS)
Modulation	GFSK
Power Consumption	~0.01x to 0.5x of reference (depending on use case)
Data Rate	LE 2M PHY: 2 Mb/s LE 1M PHY: 1 Mb/s LE Coded PHY (S=2): 500 Kb/s LE Coded PHY (S=8): 125 Kb/s
Max Tx Power	Class 1: 100 mW (+20 dBm) Class 1.5: 10 mW (+10 dbm) Class 2: 2.5 mW (+4 dBm) Class 3: 1 mW (0 dBm)
Network Topologies	Point-to-Point (including piconet), Broadcast and Mesh

Table 1: The specifications of BLE [12].

The BLE uses 40 channels spaced at 2 MHz apart. Comparing with Wi-Fi, as shown in Figure 2, BLE has more channels and smaller separation with the same frequency width [13].

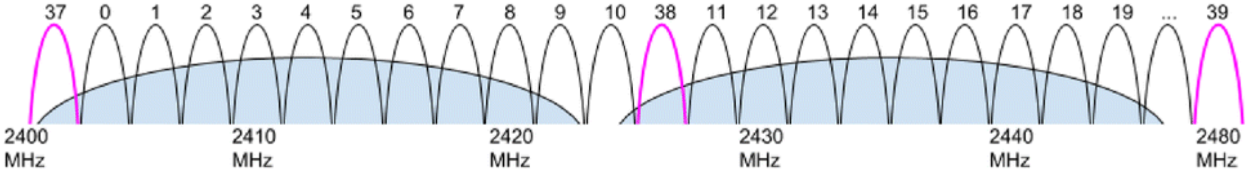


Figure 2: Comparison of BLE channels and Wi-Fi channel 1 and 6 [13].

2.1.3 Ultra-Wideband technology.

Ultra-wideband (UWB) is a wireless technology where signals are transmitted in ultra short-pulses (time period of < 1ns) and in a wide frequency band (>500 MHz), in the frequency range from 3.1 to 10.6GHz, using a very low energy level [14]. Because of using short-pulses with large frequency to transmit data, UWB signals are less affected by reflections and they can pass through various materials, including walls. Thus, UWB is a particularly attractive technology for IPSs, with high accuracy and low power consumption. However, high equipment costs and installation are disadvantages of the system using this technology [2].

2.2 Methods of position estimation.

The measured signals at the location of device can be used to determine the location of that device. To achieve this goal, we need methods of position estimation. The data for the position estimation are usually the parameters obtained from the wireless signal transmission between a source and a receiver. Commonly, they are classified into power-based, angle-based and time-based measure. Each measurement result is unique and positioning system can incorporate many types of positioning measurement to locate the device [15].

2.2.1 Time of Arrival (ToA).

Time of Arrival (ToA), sometimes called time of flight (ToF), is a method of estimating position by measuring the time interval from when the radio signal is sent from the source until the signal is received at the receiving device. Then the distance from the source to the receiver can be estimated simply by multiplying the propagation time of the signal by the speed of light. When there is at least three distances from the device to the reference point as the source, it is easy to determine the exact position of the device by finding the intersection point between the three circles with the center and the known radius, as described in figure below [10].

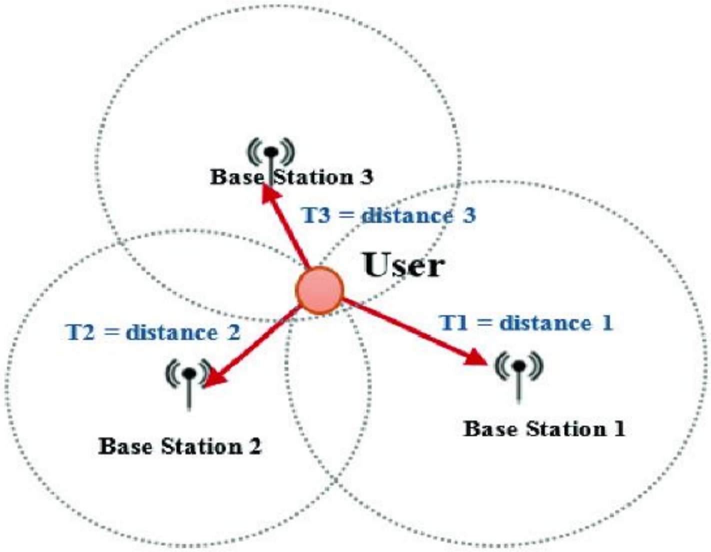


Figure 3: Time of arrival (ToA) concept [10].

2.2.2 Time Difference of Arrival (TDoA).

The TDoA method, similar to ToA, relies on the time that signals are transmitted between a transmitter and a receiver to estimate the distance. It measures the difference time of arrival of the signal sent out by multiple base stations. *Figure 4* below shows that two TDoAs (L1 and L2) are created from three base stations, the crossing point between L1 and L2 is estimated as the location of the object [10].

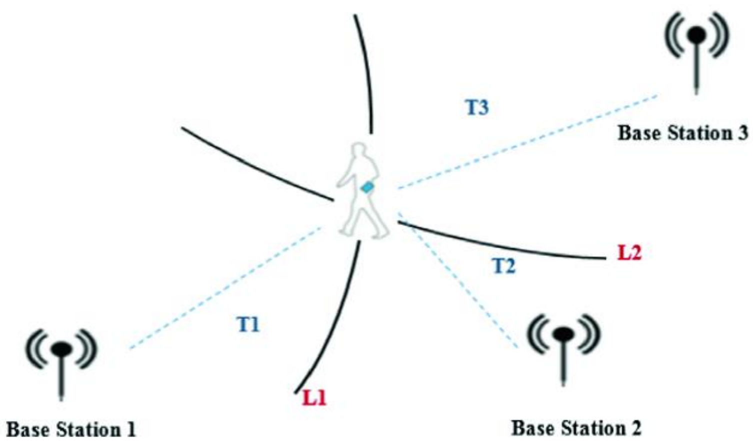


Figure 4: Time difference of arrival (TDoA) concept [10].

2.2.3 Angle of Arrival (AoA).

Angle of Arrival (AOA) method provides a calculation of the angle from the object to two or more given reference points. The location of the object can be evaluated by calculating the angle between the signal path from the source to the device and the given measurement line.

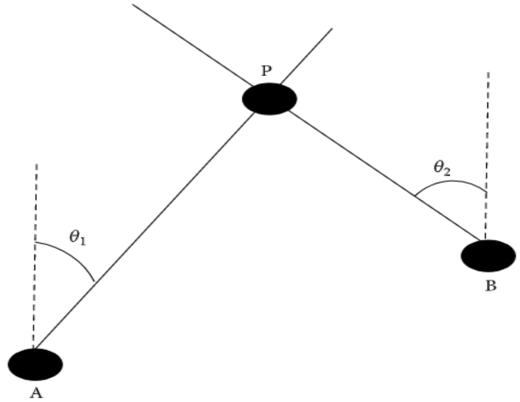


Figure 5: AoA measure [1]

2.2.4 Received Signal Strength Indicator (RSSI).

RSSI is one of the simplest and popular methods for indoor positioning. Unlike the above methods, the RSSI measures the transmitted signal strength between the object point that needs positioning and location markers. The signal strength obtained at the receiver is the remaining amount of energy after the propagation in space and partial attenuation. This method calculates the attenuation of the signal as it travels through space to determine the distance traveled by the signal, and find the relative range between the device and the reference points to locate the device. *Figure 6* below shows the RSSI based localization method [14].

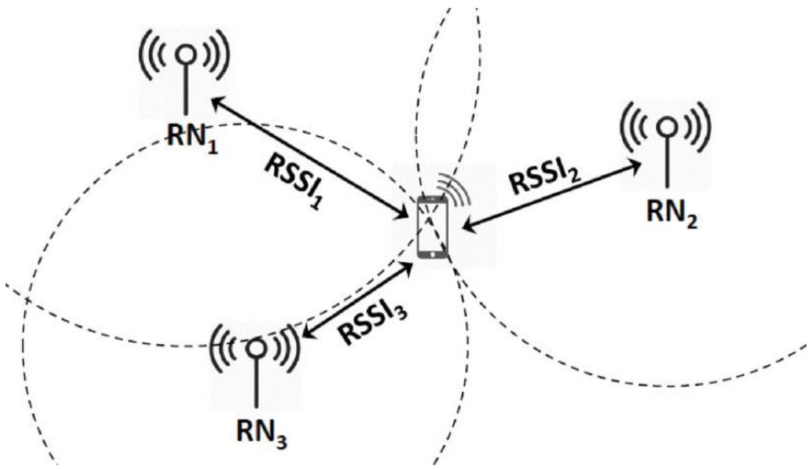


Figure 6: RSSI based localization [14].

2.3 Location techniques.

2.3.1 Fingerprinting.

Fingerprinting is one of the most widespread location technique around the world that is commonly used in an indoor positioning system. This technique is cheaper than the others as it does not require extra hardware [16]. Instead of relying on signal strength to calculate the distance between the user and the reference point, fingerprinting method utilizes the available wireless signal in the building to build a position mapping database. The process of building fingerprinting data is also known as the off-line phase. Then, the system match the real-time signal and the fingerprint database to accomplish positioning in the on-line phase [17]. The two phases of fingerprinting method is described in *Figure 7* below.

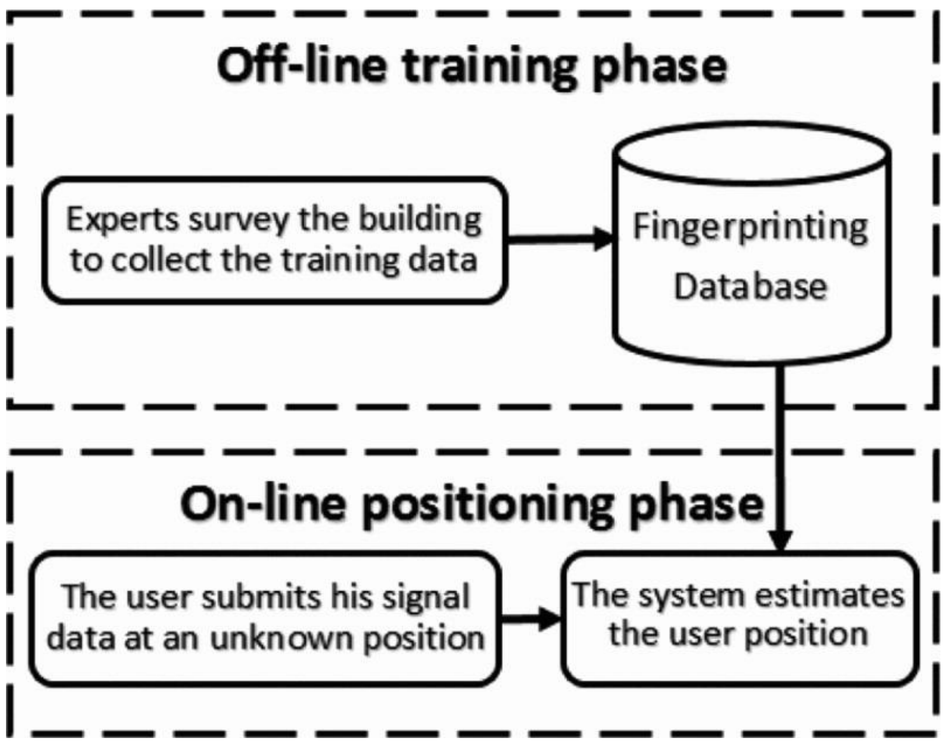


Figure 7: The two phases of fingerprinting [18].

The first phase is also called the training or planning phase. This phase measures the signal propagates inside the area to create a fingerprinting database. A wireless device, such as laptop or smart phone, moves around the area to store information the RSS at different training positions. The more training positions, the higher the accuracy of the system. The second phase is named as the estimation or positioning phase. This phase determines the location of user by comparing the current observed signal values with the signal values in the training database that have been mapped [18]. Up to now, fingerprinting based on Wi-Fi and BLE are the most popular indoor positioning methods.

2.3.1.1 Fingerprinting based on Wi-Fi.

The advantages of fingerprinting positioning technology using Wi-Fi are high accuracy and low cost. Due to Wi-Fi is being used all over the world, this method can be practiced in anywhere Wi-Fi network is available without supplementary hardware [16]. The model of Wi-Fi fingerprinting positioning technology is shown in Figure 8 below.

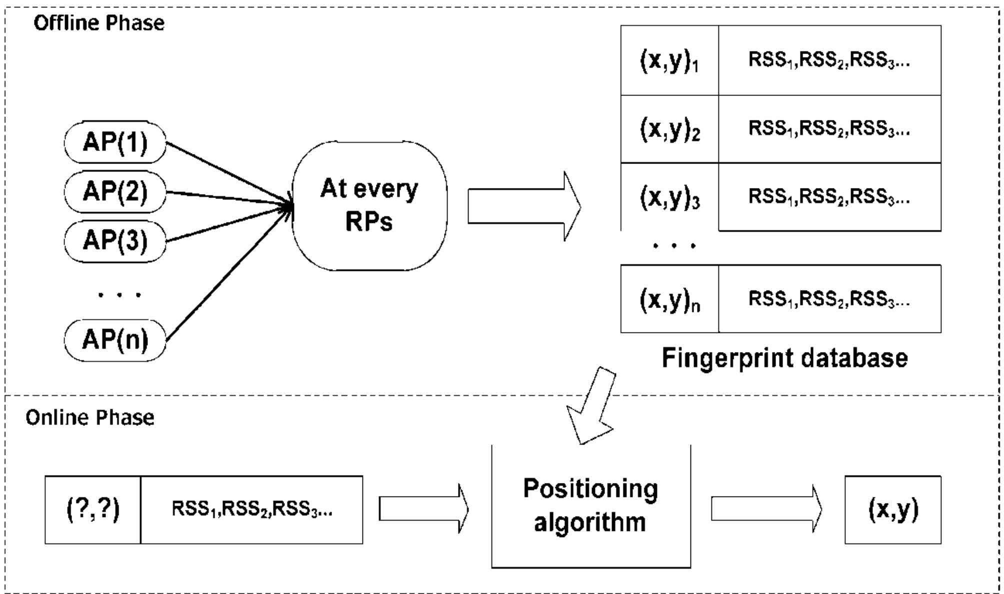


Figure 8: The model of Wi-Fi fingerprinting positioning technology [16].

In offline phase, a fingerprint database of the specific area, such as building, is grown. Firstly, the area is split up into small geographical sectors. Wi-Fi Access Points (APs) are installed at different positions in the building. Then, RSS is determined from all APs at each sector one by one and save in a database. Each sector is expressed by an array including n RSS value if this place is covered by n APs. In online phase, online RSS value of object is reported to a server which holds the fingerprint database. The system looks up the database to find the closest match RSS value with online RSS value. The actual position of object is estimated based on coordinate of this RSS value. The following formula gives the mean square error of the model with n APs [19].

$$Error = \frac{1}{n}\sum_{i=1}^{n}(RSS_{Offline}^{i} - RSS_{online}^{i})^2$$

2.3.1.2 Fingerprinting based on BLE.

Indoor localisation systems based on BLE fingerprinting use the BLE beacons to broadcasts the advertisement message. Because the transmission range of the BLE devices is shorter than transmission range of Wi-Fi devices, more BLE beacons are needed than Wi-Fi APs on the same area [20].

BLE fingerprinting method is also divided into the offline phase (training phase) and the online phase (testing phase) [4]. In the first stage, RSSIs are collected from all beacons and recorded in a fingerprint database along with their location coordinates. In online stage, the measured RSS value is matched to the values on the database to estimate the position of the object [21]. The process of BLE fingerprinting positioning technology is shown in *Figure 9* below.

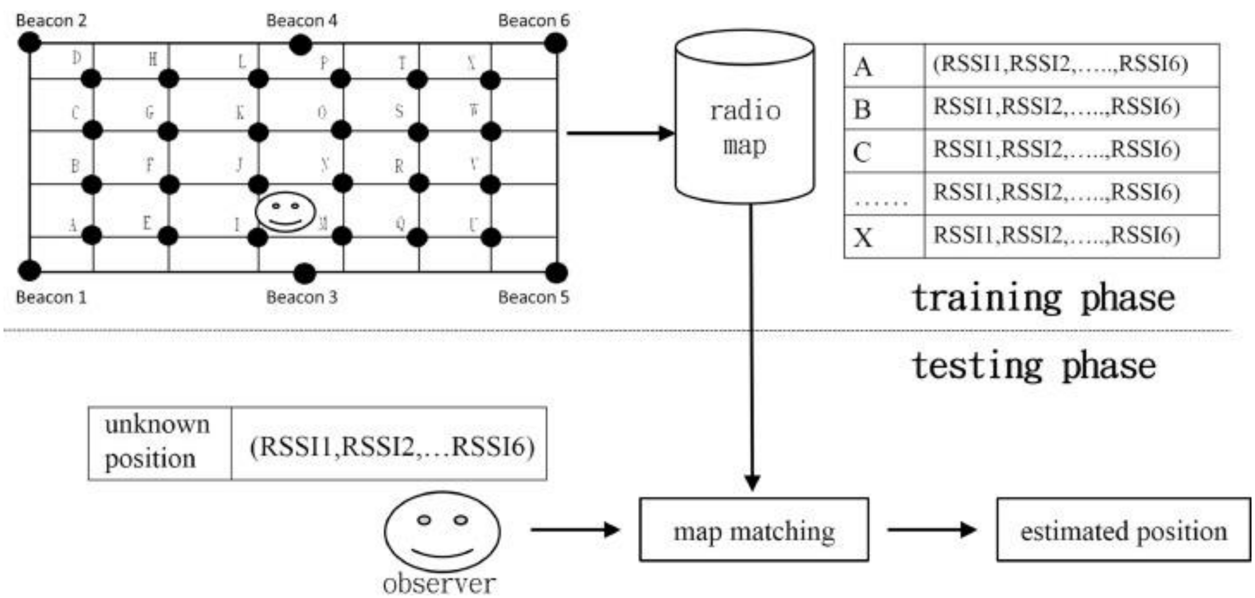


Figure 9: The model of BLE fingerprinting positioning technology [20].

2.3.2 Trilateration.

Trilateration is also one of the popular indoor location techniques. This technique based on the triangle geometry to determine the position of an object [22]. Trilateration is essentially a distance measurement between an object and three known reference points. The various position estimating methods such as RSS, ToA, TDoA and etc. are used to measure the distances. After calculating these distances, the location of the object is determined by the intersection of the three circles with a center as reference point and radius as measured distance (*Figure 10*). Although there is not an offline phase like fingerprinting technique, trilateration still needs to store the coordinates location of APs and its Mac address in a concentrated database. Based on received signal strengths from all existing APs, system calculates and then convert into distances [23].

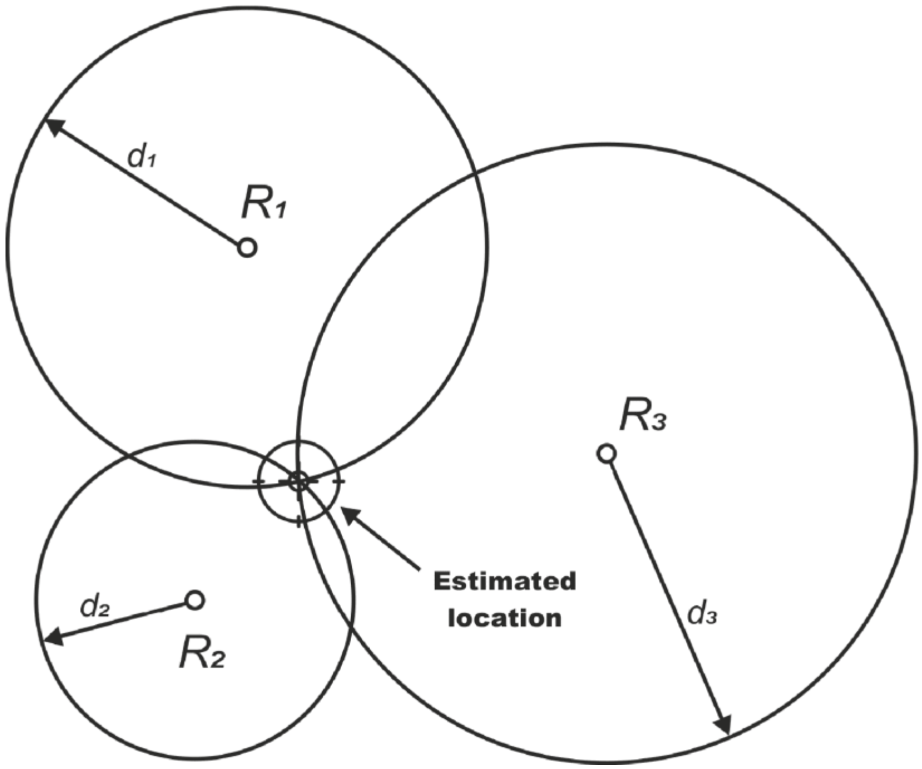


Figure 10: Position estimation in trilateration [22].

2.4 Machine Learning Fingerprinting Approaches.

"Machine Learning is a process of building computer systems that automatically improve with experience, and implement a learning process" [24]. Data mining is one of the most important applications of Machine Learning. The efficiency of systems and the designs of machines are powerfully improved through applying Machine Learning [25].

2.4.1 Types of machine learning.

Based on the design of the training database, there are two common technique types of machine learning, including [26]:

- Supervised learning: This method requires all training examples to have both the object and the label associated with it. This type of machine learning is quite common in classification problems, the process of supervised learning is displayed in the figure below:

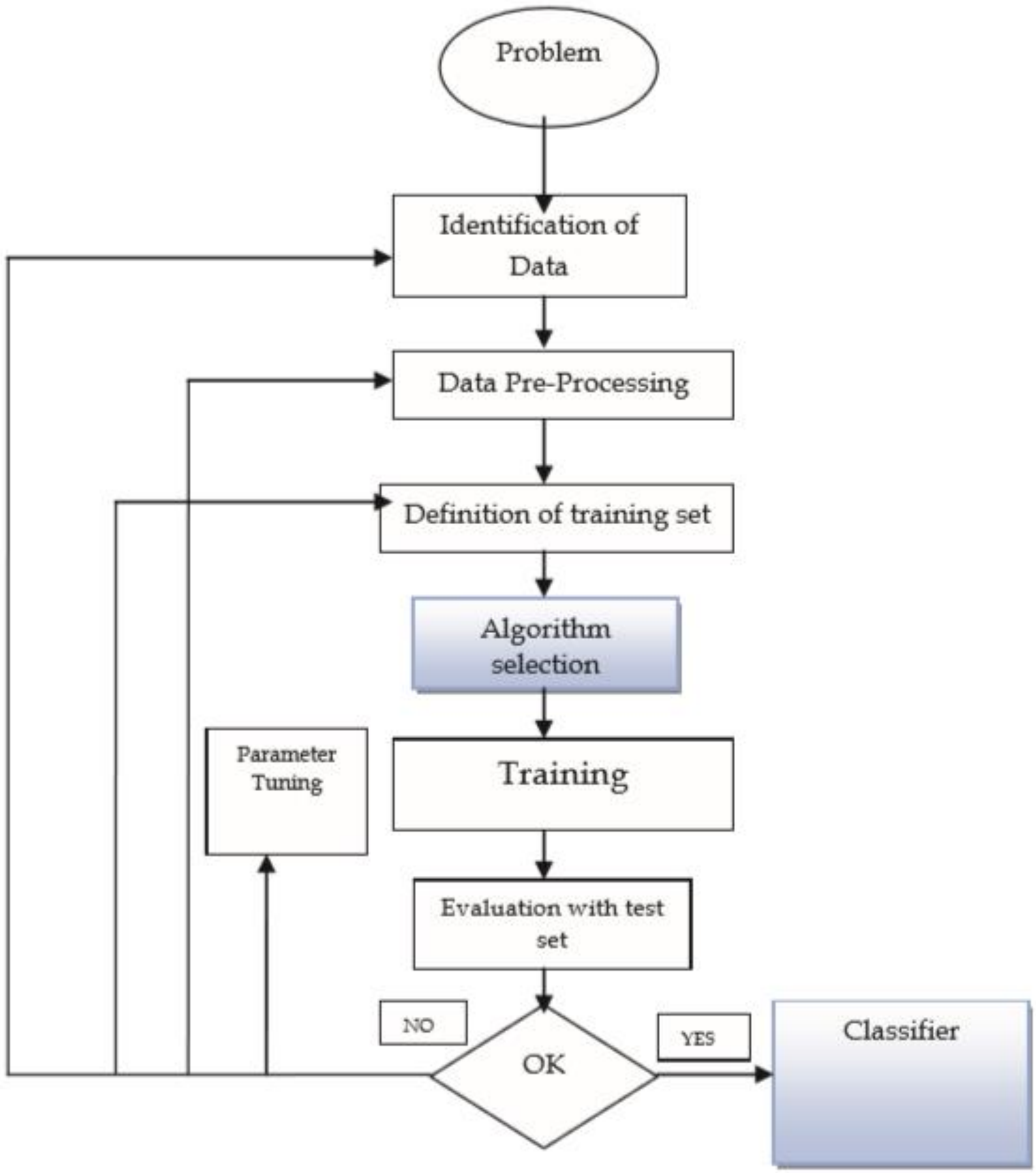


Figure 11: The process of supervised learning [27].

- Unsupervised learning: This model deals with non-labeled training examples.

With fingerprinting, supervised learning method is suitable since the training database has both the RSS value and the associated location label. In this method, there are two main approaches based on the type of the training labels, which are classification techniques and regression techniques, both of them can be used for fingerprinting [28]. While the classification techniques predict discrete responses, the regression techniques predict continuous responses. Common algorithms used for the supervised learning include support vector machine, discriminant analysis, Naive Bayes, nearest neighbor, linear regression, ensemble methods, decision trees, and neural networks [29]. Following *Figure 12* describes the machine learning techniques and common algorithms.

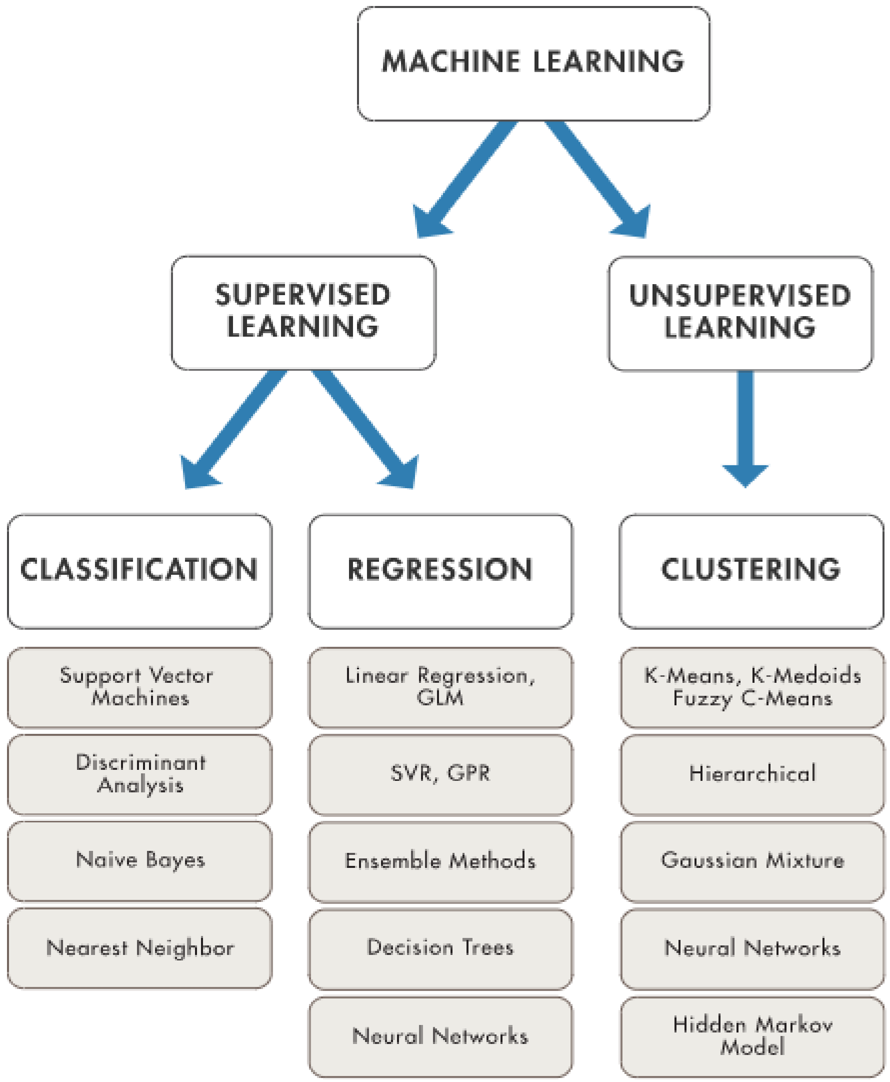


Figure 12: Machine learning techniques and common algorithms [29].

The next sections mention to two of the algorithms of supervised learning method as Nearest Neighbors and Naive Bayes.

2.4.2 Classification fingerprinting with K-nearest neighbours.

K-Nearest Neighbors (KNN) algorithm is the deterministic algorithm category. This is one of the simplest supervised learning algorithms in Machine Learning to estimate radio-based indoor position [30].

The training database, including reference points (Bi position and Ri vector of the RSSI, called the fingerprint) is stored in the offline phase (or training phase). KNN checks several most akin neighbors with fingerprints during the online phase. The signals that the transmitter gains from an unknown location are compared to the data stored in the fingerprint database. A Ri fingerprint that best matches the received signal is estimated for the location. However, the accurate position is not likely to happen. Thus, the k most similar fingerprints are chosen, and the mean of coordinates of these fingerprints is the estimated position [31].

The fingerprint vector form of the i^{th} reference point (a) and a detail of the fingerprinting database with (x, y) coordinate (b) are displayed below [32]:

$$\overrightarrow{RSS}_{i} = [RSS_{i,1}, RSS_{i,2}, \ldots, RSS_{i,n}] \text{ (a)}$$

$$\overrightarrow{RSS}_{DBi} = [x_i, y_i \mid RSS_{i,1}, RSS_{i,2}, \ldots, RSS_{i,n}] \text{ (b)}$$

where,

$RSS_{i,j}$: RSS value of j^{th} transmitter determined at i^{th} reference point

n: the number of transmitters.

The distance between two fingerprints simply calculates using Euclidian metric formula:

$$L_{m,i} = \sqrt{\left(\sum_{j-1}^{n} |RSS_{m,j} - RSS_{DBi,j}|^2\right)}$$

where, $L_{m,i}$ is the distance between fingerprint acquired at unknown point designated with m, and i^{th} RP from the database. After using above formula, the k matches are defined. Then, the coordinate of object is determined by averaged value of k neighbours, as formulated below [32]:

$$x_m = \frac{\sum_{i=1}^{k} x_i}{k}, \qquad y_m = \frac{\sum_{i=1}^{k} y_i}{k}$$

2.4.3 Classification fingerprinting with Naive Bayes.

Naive Bayes (NB) is the probabilistic algorithm category. This is a widely used supervised learning method in fingerprinting technique [33]. While KNN algorithm predict an unknown point place by computing an average of all nearest locations, NB algorithm choose the highest probability position in the fingerprinting database to show an unknown point.

Suppose that $\overrightarrow{RSS_u}$ is the actual RSS vector at an unknown point, NB method computes the posterior probability $P(L_i|\overrightarrow{RSS_u})$ of this vector being observed at the training point L_i ($1 \leq i \leq M$). The estimated point is chosen that is the training point with $\max(P(L_i|\overrightarrow{RSS_u}))$. This probability is determined by Bayes' theory as follows [34]:

$$P(L_i|\overrightarrow{RSS_u}) = \frac{P(L_i)P(\overrightarrow{RSS_u}|L_i)}{P(\overrightarrow{RSS_u})} \quad (*)$$

In above (*) formula, $P(\overrightarrow{RSS_u})$ and $P(L_i)$ are known. The $P(\overrightarrow{RSS_u})$ is a constant and is independent with the location variable L. $P(L_i)$ is the prior probability of the object being at position L_i when the measurements are unknown.

We need to calculate $P(\overrightarrow{RSS_u}|L_i|)$. By assuming all APs are independent, NB method determine the $P(\overrightarrow{RSS_u}|L_i|)$ as follows:

$$P(\overrightarrow{RSS_u}|L_i|) = \prod_{j=1}^{N} P(AP_j^u|L_i)$$

where, $P(AP_j^u|L_i) = \dfrac{number\ of\ times\ APj\ appears\ at\ location\ Li}{total\ number\ of\ readings\ observed\ at\ location\ Li}$

CHAPTER 3: PROJECT PLAN

3.1 Objective.

This thesis evaluates the performance of Wi-Fi Fingerprinting based Indoor Positioning System. Based on the existing access points, a radio map of the area of interest is built, including coordinates of reference points and related RSS values. If this area has x APs, each reference point is expressed by x RSS values. In an off-line phase, the laptop connect with SparkFun ESP32 Thing board will be used to collect the fingerprints in the selected area. This data is used to train a simple machine learning model. In an on-line phase, the device to be located will automatically send measured signal strength data to the localisation server. The server will use the pre-trained classifier to track the location of the device. The final step in the project will involve evaluation of the system's performance.

In order to accomplish the above objective, firstly, the area covered by multiple APs will be selected for experiment. At the data collection stage, the SparkFun ESP32 Thing board scans the data at the Reference Points (RPs) in turn and outputs the files with extension .log via PuTTY software. Then, a MATLAB program is used to convert and combine these data into the file in CSV format, making the input data for the next steps. The Machine Learning App with MATLAB is used to perform data analysis and evaluation steps.

3.2 Experimental areas.

SSID	MAC Address	RSSI	Chan	Max Speed	WEP	WPA	WPA2	WPS	Vendor
unisa-help	58:AC:78:9C:B7:F4	-52	6	216.7 Mbps	Open				Cisco Systems. Inc
Ignition	58:AC:78:9C:B7:F2	-53	6	216.7 Mbps			PSK-CCMP		Cisco Systems. Inc
Ignition	58:AC:78:9C:B7:FD	-55	64+60	1300.05 Mbps			PSK-CCMP		Cisco Systems. Inc
eduroam	58:AC:78:9C:B7:FE	-55	64+60	1300.05 Mbps			MGT-CCMP		Cisco Systems. Inc
unisa-help	58:AC:78:9C:B7:FB	-55	64+60	1300.05 Mbps	Open				Cisco Systems. Inc
unisa	58:AC:78:89:EE:E0	-60	11	216.7 Mbps			MGT-CCMP		Cisco Systems. Inc
Ignition	58:AC:78:89:EE:E2	-60	11	216.7 Mbps			PSK-CCMP		Cisco Systems. Inc
eduroam	58:AC:78:89:EE:E1	-60	11	216.7 Mbps			MGT-CCMP		Cisco Systems. Inc
unisa-help	58:AC:78:89:EE:E4	-60	11	216.7 Mbps	Open				Cisco Systems. Inc
unisa-guest	58:AC:78:89:EE:E3	-60	11	216.7 Mbps			MGT-CCMP		Cisco Systems. Inc
unisa-help	84:B2:61:D1:29:FB	-62	48+44	1300.05 Mbps	Open				Cisco Systems. Inc
unisa-guest	58:AC:78:89:EE:EC	-62	149+153	1300.05 Mbps			MGT-CCMP		Cisco Systems. Inc
unisa	84:B2:61:D1:29:FF	-63	48+44	1300.05 Mbps			MGT-CCMP		Cisco Systems. Inc
eduroam	84:B2:61:D1:29:FE	-63	48+44	1300.05 Mbps			MGT-CCMP		Cisco Systems. Inc
unisa-guest	84:B2:61:D1:29:FD	-63	48+44	1300.05 Mbps			MGT-CCMP		Cisco Systems. Inc
unisa	58:AC:78:89:EE:EF	-63	149+153	1300.05 Mbps			MGT-CCMP		Cisco Systems. Inc
eduroam	58:AC:78:89:EE:EE	-63	149+153	1300.05 Mbps			MGT-CCMP		Cisco Systems. Inc
unisa-help	58:AC:78:89:EE:EB	-63	149+153	1300.05 Mbps	Open				Cisco Systems. Inc
unisa-help	18:80:90:B4:CC:04	-73	11	216.7 Mbps	Open				Cisco Systems. Inc
unisa	18:80:90:B4:CC:00	-73	11	216.7 Mbps			MGT-CCMP		Cisco Systems. Inc
unisa-guest	84:B2:61:D1:29:F2	-74	1	216.7 Mbps			MGT-CCMP		Cisco Systems. Inc
unisa	84:B2:61:D1:29:F0	-74	1	216.7 Mbps			MGT-CCMP		Cisco Systems. Inc
unisa	58:AC:78:A3:99:50	-74	6	216.7 Mbps			MGT-CCMP		Cisco Systems. Inc

Figure 13: The coverage of Wi-Fi APs in experimental area.

The chosen area for experiment is a part of the first floor of the SCT building, Mawson Lakes campus. This place is covered by more than 50 APs that are located within 100m. The *Figure 13* lists some Wi-Fi APs in experimental area.

The selected location is divided the 3m x 3m grid and RPs are located at the crossings of the grid in the corridors and Electronics Innovation Laboratory of the building. The area of experimental space is around 350 m2. The number and location of the RPs are described in the *Figure 14* below.

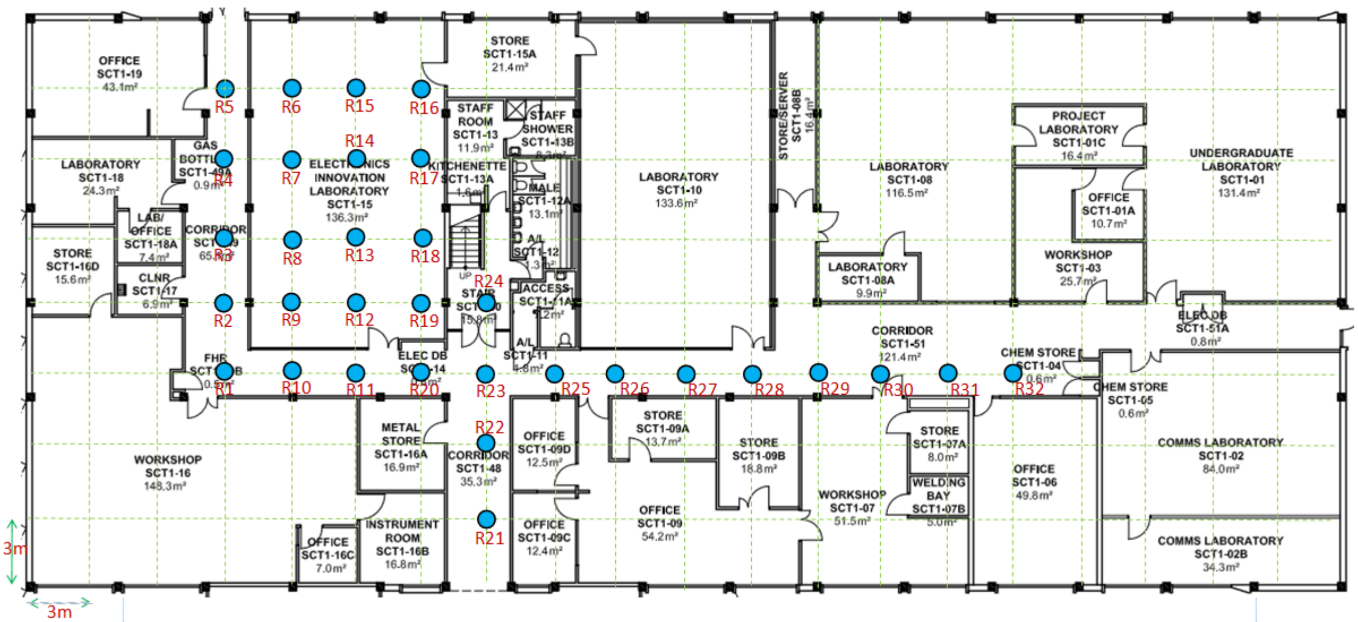


Figure 14: Location of reference points in STC Building.

3.3 Hardware implementation.

The fingerprinting data collection and process involve the main hardware as laptop and SparkFun ESP32 Thing board. The APs are the ones available in the selected area.

3.3.1 SparkFun ESP32 Thing.

One of the famous IoT device platforms today is the SparkFun ESP32 Thing. This is a Wi-Fi and BLE compatible microcontroller, where the ESP32 is equipped with everything necessary to run and program. It can be used as both receiver and transmitter [35]. Pin Layout and some specifications of SparkFun ESP32 Thing are shown in *Figure 15* and *Table 2* below.

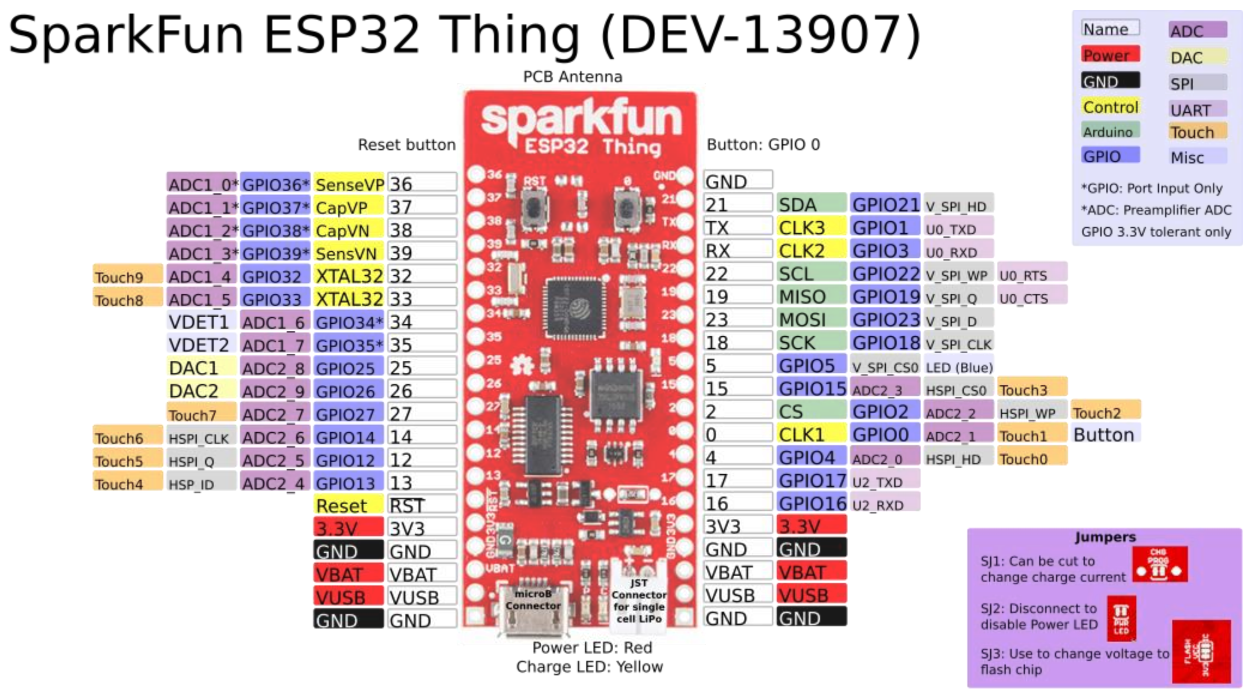


Figure 15: : SparkFun ESP32 Thing Pin Layout [36].

Categories	**Items**	**Specifications**
	Power supply	2.2 V ~ 3.6 V
	Output of regular	3.3V/600mA
	Sleep current under hibernation	2.5 µA
	Module interfaces	3 SPI interfaces, 3 UART interfaces, 2 I^2C interfaces, 16 PWM outputs, 2 I^2S, 18 ADC channels, 2 DACs, 10 capacitive touch inputs.
	Internal SRAM	520 kB
	External flash	4 MB
Wi-Fi	Protocols	802.11 b/g/n/e/i
		A-MPDU and A-MSDU aggregation
	Security protocols	WPA/WPA2/WPA2-Enterprise/WPS

Bluetooth	Protocols	Bluetooth v4.2 BR/EDR and BLE specification
	Radio	NZIF receiver with –98 dBm sensitivity
		Class-1, class-2 and class-3 transmitter without external power amplifier

Table 2: Some specifications of SparkFun ESP32 Thing [36].

3.3.2 The Laptop.

A laptop is used to communicate with SparkFun ESP32 Thing board via a USB extension cable to collect and save fingerprinting database.

3.4 Software preparation.

3.4.1 Arduino IDE.

The Arduino Integrated Development Environment (IDE) is a open-source software that is used to write code and upload it to Arduino compatible boards, or developed boards by other supported manufacturers. This software is a multi-platform application that is written in Java and can operates on Windows, Mac OS X, and Linux [37].

In this project, Arduino IDE will be used to upload a program (also called sketch, and file format with extension .ino) to the ESP32 board for scanning neighborhood Wi-Fi APs. The output will be SSIDs (service set identifier) and RSSIs of APs. The sketch is created base on the Wi-Fi library in Arduino IDE, code of program begin with: *#include <WiFi .h>*. Some main syntaxes are used for above purpose as below table:

Syntax	Description	Returns
WiFi.scanNetworks()	Scan for nearby networks	Number of discovered networks
WiFi.BSSID()	Gets the MAC address of the router you are connected to	A byte array containing the MAC address of the WiFi router is currently connected to
WiFi.RSSI()	Obtains the signal strength of the connection to the router	The current RSSI in dBm

Table 3: Some main syntaxes in the Wi-Fi library in Arduino IDE [38].

3.4.2 PuTTY Software.

PuTTY is open source software, created by Simon Tatham, a British computer programmer. This software supports various network protocols such as Telnet, SSH (Secure Shell), Rlogin, etc. as well as can connect to a serial port [39].

In this project, PuTTY software is used in the fingerprinting collection step. A computer connects to the ESP-32 board via USB port to scan information of APs, at the same time, be connected to PuTTY by the serial line to get the collected data from ESP. PuTTY configuration for this task is described in *Figure 16* below. The main parts include choosing the serial line and selecting the its speed. In this case, the serial line is COM3 and the speed is 15200 which is the same with the speed in the sketch uploaded in the ESP-32.

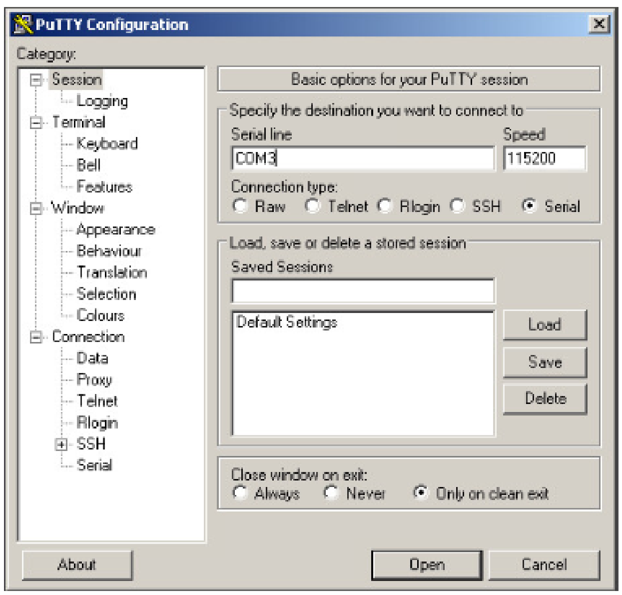


Figure 16: PuTTY configuration.

3.4.3 MATLAB.

MATLAB, which stands for Matrix Laboratory, is a programming platform designed by MathWorks corporation. With the apps, language and built-in math functions, MATLAB is used to analyze data, develop algorithms, and create models and applications. MATLAB can interface with other programming languages such as Java, C, C ++, and Fortran to deploy applications in many various fields, including signal processing and imaging, communications, control design, test measurement, financial model analysis, or biological computation [40].

The MATLAB used in this project has two purposes. Firstly, a program is built to convert the files which are created from the ESP-32 board into an appropriate data format (Excel file). Then, using Machine Learning Apps with MATLAB to analyze and evaluate data. Classification Learner App and Regression Learner App in the MATLAB R2020a version are used for this experiment.

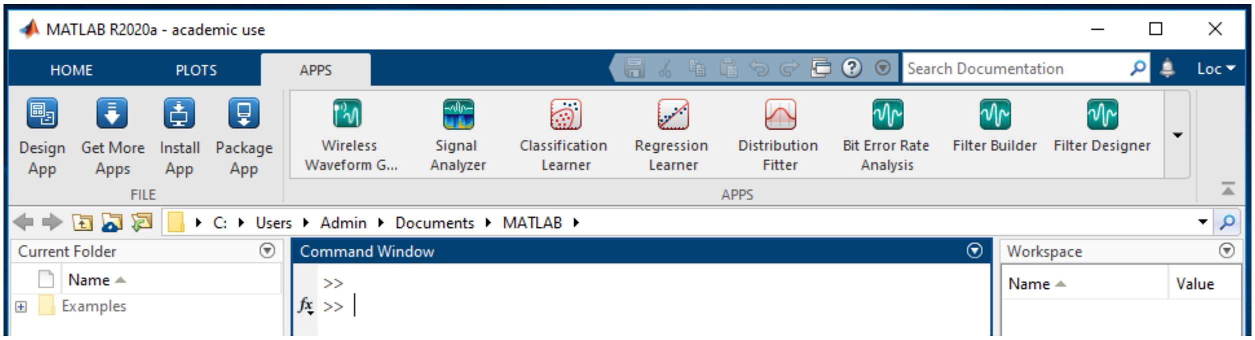


Figure 17: Interface of the MATLAB R2020a version.

The common workflow of training classification models and regression models in corresponding learner apps are shown in following figures.

Figure 18: Workflow of classification models in the Classification Learner app [41].

Figure 19: Workflow of regression models in the Regression Learner app [41].

The Machine Learning Apps with MATLAB allow both automated and manual classifier training. With Classification Learner app, classifier types include decision trees, support vector machines, nearest neighbor classifiers, Naive Bayes classifiers, ensemble classifiers, discriminant analysis, and logistic regression. While, linear regression models, regression trees, support vector machines, Gaussian process regression models, and ensembles of trees are the regression model types in Regression Learner app [41].

3.5 Time plan.

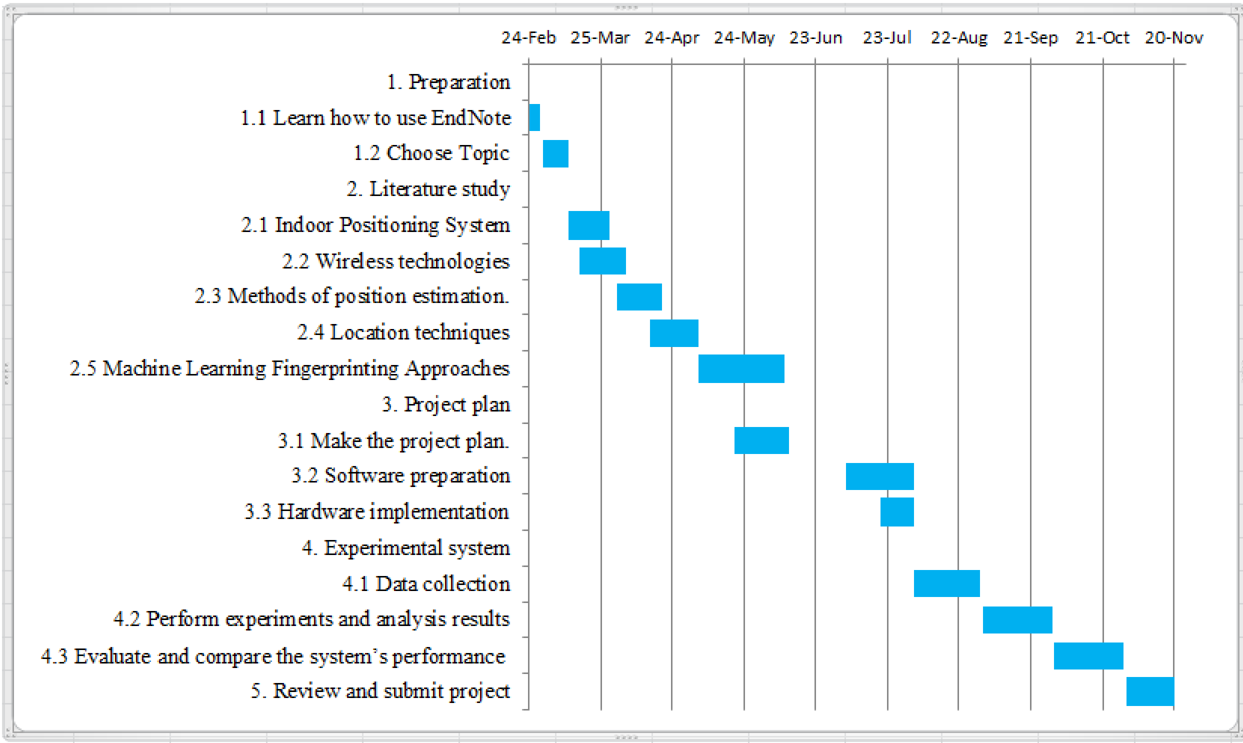


Table 4: Time plan of the project.

CHAPTER 4: EXPERIMENTAL SYSTEM

4.1 Data collection process.

4.1.1 Scan the data at the RPs.

In this stage, the Sparkfun ESP32 board is used for data scanning. It scans the data at the RPs in turn and outputs the files with extension .log. The data output is the MAC address and RSSI of the APs identified by the ESP32 board. There are 40 scans with 5 seconds delay for each time are executed at each RP. Data is retrieved manually by carrying the device to move through each position one by one. This was done on 31/08/2020 at SCT building. The sketch is used to upload to the ESP32 board (via Arduino IDE) as below.

Scan_Wifi §

```
#include "WiFi.h"
void setup()
{
    Serial.begin(115200);
    // Set WiFi to station mode and disconnect from an AP if it was previously connected
    WiFi.mode(WIFI_STA);
    WiFi.disconnect();
    delay(100);
    Serial.println("Setup done");
}
void loop()
{
    Serial.println("scan start");
    // WiFi.scanNetworks will return the number of networks found
    int n = WiFi.scanNetworks();
    Serial.println("scan done");
    if (n == 0) {
        Serial.println("no networks found");
    } else {
        Serial.print(n);
        Serial.println(" networks found");
        for (int i = 0; i < n; ++i) {
            // Print SSID and RSSI for each network found
            Serial.print(i + 1);
            Serial.print(": ");
            Serial.print(WiFi.BSSIDstr(i));ss
            Serial.print(" (");
            Serial.print(WiFi.RSSI(i));
            Serial.print(")");
            Serial.println((WiFi.encryptionType(i) == WIFI_AUTH_OPEN)?" ":"*");
            delay(10);
        }
    }
    Serial.println("");
    // Wait a bit before scanning again
    delay(5000);
}
```

Figure 20: The scanning sketch uploaded in ESP 32 board.

The output data format via PuTTY in this step is shown in following *Figure 21*. MAC addresses and RSSI values of all APs are displayed in the same row for each scan in .log file of each RP.

```
SCT_R01_3108 - Notepad
File  Edit  Format  View  Help
=~=~=~=~=~=~=~=~=~=~=~= PuTTY log 2020.08.31 13:23:35 =~=~=~=~=~=~=~=~=~=~=~=
58:AC:78:89:EE:E0,-49,58:AC:78:89:EE:E4,-49,58:AC:78:89:EE:E2,-49,58:AC:78:89:EE:E1,-49,58:AC:78:89:EE:E3,-49,!
58:AC:78:89:EE:E0,-50,58:AC:78:89:EE:E4,-50,58:AC:78:89:EE:E2,-50,58:AC:78:89:EE:E1,-50,58:AC:78:89:EE:E3,-50,!
58:AC:78:89:EE:E0,-56,58:AC:78:89:EE:E4,-56,58:AC:78:89:EE:E2,-56,58:AC:78:89:EE:E1,-56,58:AC:78:89:EE:E3,-56,!
58:AC:78:89:EE:E0,-54,58:AC:78:89:EE:E4,-54,58:AC:78:89:EE:E2,-54,58:AC:78:89:EE:E1,-54,58:AC:78:89:EE:E3,-54,!
58:AC:78:89:EE:E0,-53,58:AC:78:89:EE:E4,-53,58:AC:78:89:EE:E2,-53,58:AC:78:89:EE:E1,-53,58:AC:78:89:EE:E3,-53,!
58:AC:78:89:EE:E0,-50,58:AC:78:89:EE:E4,-50,58:AC:78:89:EE:E2,-50,58:AC:78:89:EE:E1,-50,58:AC:78:89:EE:E3,-51,!
58:AC:78:89:EE:E0,-53,58:AC:78:89:EE:E2,-53,58:AC:78:89:EE:E3,-53,58:AC:78:89:EE:E4,-54,58:AC:78:89:EE:E1,-54,!
```

Figure 21: The output data format via PuTTY.

4.1.2 Convert data to the required format.

In this stage, the scanned records from all RPs are aggregated into a large file having an appropriate format for data analysis. Firstly, the log file in each RP is changed to the files with extension .csv. It is then combined into a common file containing data of all RPs. Finally, a Matlab program is used to convert above file to the file in the required format (as the *Figure 22* below). In the output file, the value -110 is assigned to the MAC addresses that are not detected during the scan. Matlab code is used as below.

```
clear;
table = readtable('Logfile_csv.csv') ; % load csv file
[A,B] = size(table);

labels = [];
for j=1:A % gather MAC addresses
    fprintf('Gathering MAC addresses: row:%d/%d\n', j, A);
    for i=2:2:B
        labels = [labels ; table{j,i}];
    end
end
labels = labels(~cellfun('isempty',labels));
labels = unique(labels, 'stable'); % no. of unique MAC addresses

C = size(labels);
C = C(1);
for i=1:C
    my_matrix(1,i+1) = labels(i);
end

for k=1:A % configure table
    fprintf('Configuring table: row:%d/%d\n', k, A);
    my_matrix{k+1,1}=table{k,1};
```

```
    for i=2:2:B
        for j=1:C
            if strcmp(table{k,i},labels(j))
                if isa(table{k,i+1},'cell')
                    temp=str2double(table{k,i+1});
                else
                    temp=table{k,i+1};
                end
                my_matrix{k+1,j+1}=temp;
            end
        end
    end
end

for i=2:A+1 % assigning -110 for blank entries
    for j=2:C+1
        if isempty(my_matrix{i,j})
            my_matrix{i,j}=-110;
        end
    end
end

headers = my_matrix(1,:); % contructing the table
headers{1,1} = ' ';
m = my_matrix(2:A+1,:);
T  = cell2table(m);
T = splitvars(T);
T.Properties.VariableNames = headers;
writetable(T, 'output_collect_data.xlsx') ; % save table
```

The output of the Matlab program is an excel file of size 1281 x 57. The first row shows a list of all the MAC addresses scanned by the device, 1280 rows that follow (32 RPs and 40 scans per each RP) are the RSSI values of APs received by the device at the RPs. The format of the output file is as below.

	A	B	C	D	E	F	G
1		58:AC:78:89:EE:E0	58:AC:78:89:EE:E4	58:AC:78:89:EE:E2	58:AC:78:89:EE:E1	58:AC:78:89:EE:E3	58:AC:78:9C:B7:F0
2	RefPoint_01	-49	-49	-49	-49	-49	-65
3	RefPoint_01	-50	-50	-50	-50	-50	-69
4	RefPoint_01	-56	-56	-56	-56	-56	-68
5	RefPoint_01	-54	-54	-54	-54	-54	-69
6	RefPoint_01	-53	-53	-53	-53	-53	-64
7	RefPoint_01	-50	-50	-50	-50	-51	-67
8	RefPoint_01	-53	-54	-53	-54	-53	-70
9	RefPoint_01	-54	-54	-54	-54	-54	-70
10	RefPoint_01	-54	-54	-54	-54	-54	-70

Figure 22: The format of excel file in output of the Matlab program.

4.2 Training of the machine learning model for WiFi.

The collected data is processed by using machine learning tools in Matlab. The three apps used in this project are Classification Learner, Neural Net Pattern Recognition, and Neural Net Fitting. The details of data analysis process are shown in the parts below.

4.2.1 Classification Learner.

The Classification Learner is the app to train models via numerous classifiers, such as decision trees, naive Bayes, support vector machines, nearest neighbors, and ensemble classifiers. In Classification Learner, training a model be contained in two parts: validated model and full model. While validated model is a training with a validation scheme, full model is a training on full data without validation. The validated model is used in this experiment. This method gives a good estimate of the predictive accuracy of the final model trained with all the data. The model accuracy, plots, or confusion matrix chart are reflected in the validated model results.

The output file of the Matlab program in above step (4.1.2) is the data set used for classification learner. The cross-validation is the selected validation scheme. The results achieved after training are as follows.

4.2.1.1 Check performance.

No.	Model Type	Accuracy	Machine Learning Technique
Model 1.1	Fine Tree	88.9%	DECISION TREES (DTs)
Model 1.2	Medium Tree	45.5%	
Model 1.3	Coarse Tree	15.2%	
Model 1.7	Kernel Naive Bayes	84.3%	NAIVE BAYES
Model 1.8	Linear SVM	89.6%	SUPPORT VECTOR MACHINES (SVM)
Model 1.9	Quadratic SVM	89.5%	
Model 1.10	Cubic SVM	88.9%	
Model 1.11	Fine Gaussian SVM	66.2%	
Model 1.12	Medium Gaussian SVM	87.0%	
Model 1.13	Coarse Gaussian SVM	79.4%	

Model 1.14	Fine KNN	81.2%	K-NEAREST NEIGHBOR (K-NN)
Model 1.15	Medium KNN	82.0%	
Model 1.16	Coarse KNN	48.1%	
Model 1.17	Cosine KNN	81.6%	
Model 1.18	Cubic KNN	79.1%	
Model 1.19	Weighted KNN	84.2%	
Model 1.20	Boosted Trees	87.0%	ENSEMBLE CLASSIFIERS
Model 1.21	***Bagged Trees***	***91.7%***	
Model 1.22	Subspace Discriminant	75.3%	
Model 1.23	Subspace KNN	88.6%	
Model 1.24	RUSBoosted Trees	45.6%	

Table 5: The accuracy percentage in model for Wi-Fi.

The overall accuracy in percent of models after training in Classification Learner are listed in *Table 5*. According to above table, Bagged Trees is the model type having the best accuracy score, with percentage of 91.7%. This score is the validation accuracy. It is used to estimates a performance of model on new data compared to the training data and select the best model.

4.2.1.2 Check performance per class.

The confusion matrix plot is used to examine performance per class of a training model. It helps identify the places having poor performed classifier. The *Figure 23* shows the confusion matrix with the number of observations of model type Bagged Trees. The true class and the predicted class are displayed by rows and columns respectively, and they are matched at the diagonal cells. The class is classified correctly when these cells are blue.

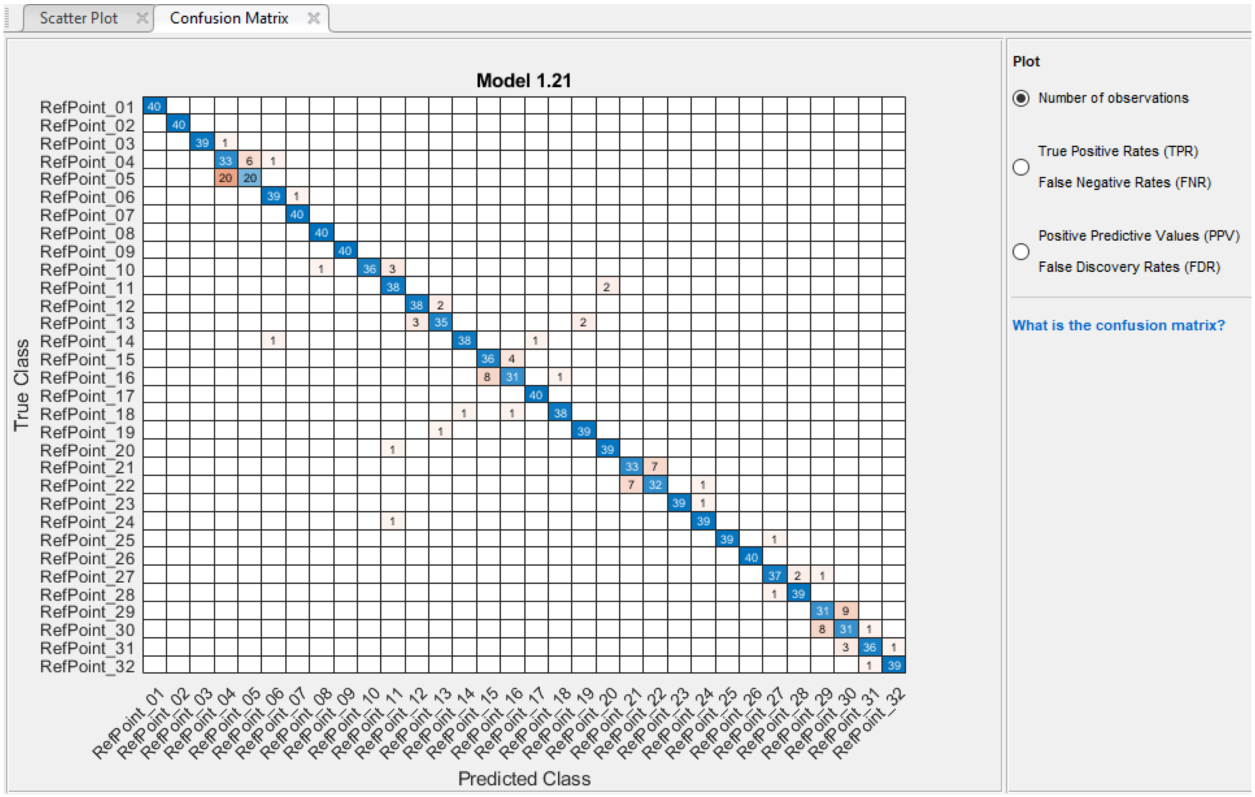


Figure 23: Confusion Matrix in model for Wi-Fi.

4.2.1.3 Check the ROC curve.

The receiver operating characteristic (ROC) curve displays the performance of a classification model after training. This graph includes two parameters: true positive rate (TPR) and false positive rate (FPR). The performance of the currently selected classifier is marked on the plot by the values of TPR and FPR. The example with Model 1.21 in *Figure 24* below, 0% of the observations incorrectly to the positive class is assigned for the current classifier when a FPR of 0.00. A TPR of 1.00 shows that the current classifier gives 100% of the observations correctly to the positive class. This is a perfect result when there is no misclassified points. The Area Under Curve (AUC) is a measure used to evaluate the overall quality of the classifier. It measures the entire area below the ROC curve from (0,0) to (1,1). The perfect classifier has an AUC value of 1.00, and the better classifier performance is indicated by a larger AUC value.

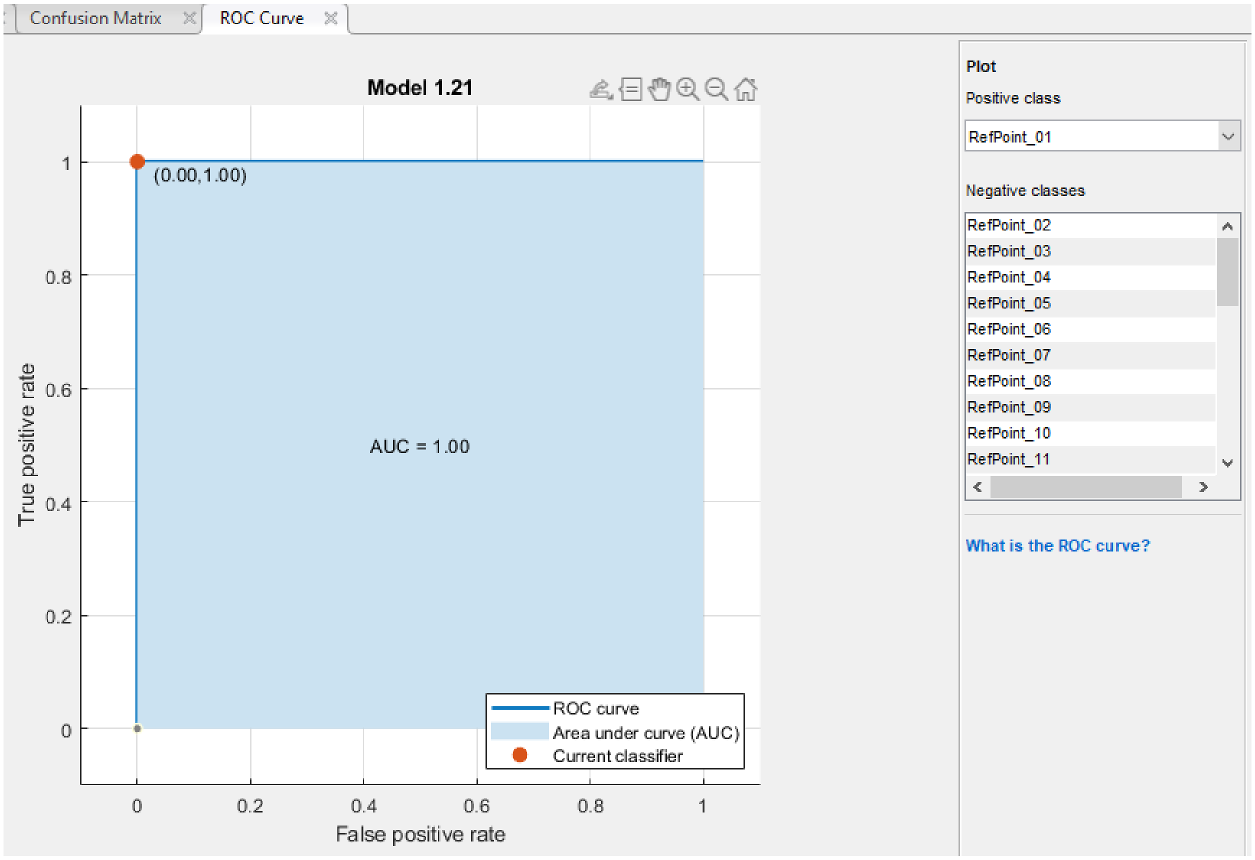


Figure 24: ROC Curve in model for Wi-Fi.

4.2.2 Neural Net Pattern Recognition.

The Neural Net (NN) Pattern Recognition app uses a two-layer feed-forward network to solve a data classification problem. It helps us select data, create and train a network. The performance is evaluated by using cross-entropy and percent misclassification error. Details of this app's operation as follows.

4.2.2.1 Select data.

The NN Pattern Recognition needs Input data and the Target data to train. The input data is a 1280x56 matrix achieved by removing the first column (RPs name) and the first row (MAC addresses) from the output file of Matlab program *(Figure 25)*. The desired network output is defined in the target data. This is a 1280x32 matrix, containing the cells are marked as one in the columns that match the RPs, and are the zero values in the elsewhere *(Figure 26)*.

	A	B	C	D	E	F	G
1	-49	-49	-49	-49	-49	-65	-65
2	-50	-50	-50	-50	-50	-69	-68
3	-56	-56	-56	-56	-56	-68	-70
4	-54	-54	-54	-54	-54	-69	-68
5	-53	-53	-53	-53	-53	-64	-63
6	-50	-50	-50	-50	-51	-67	-67
7	-53	-54	-53	-54	-53	-70	-70
8	-54	-54	-54	-54	-54	-70	-70
9	-54	-54	-54	-54	-54	-70	-69
10	-54	-54	-55	-55	-55	-69	-68

Figure 25: The format of input data in NN Pattern Recognition.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
16	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
17	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
18	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
19	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
21	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
22	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
23	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
24	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
25	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
26	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
27	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
28	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
29	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
30	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
31	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
32	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
33	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
34	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
35	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
36	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
37	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
38	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
39	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
40	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
41	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
42	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 26: The format of target data in NN Pattern Recognition.

4.2.2.2 Validation and test data.

The original data are randomly divided into three kinds:

- Training: 70%. These are presented to the network during training, and the network is adjusted according to its error.

- Validation: 15%. These are used to measure network generalization, and to halt training when generalization stops improving.

- Testing: 15%. These have no effect on training and so provide an independent measure of network performance during and after training.

The *Figure 27* below shows random division of 1280 samples in these sets.

Results

	Samples	CE	%E
Training:	896	5.31001e-0	4.57589e-0
Validation:	192	15.00388e-0	15.10416e-0
Testing:	192	15.00545e-0	14.06250e-0

Figure 27: Select percentages.

4.2.2.3 Set the network architecture.

We define the number of neurons in the pattern recognition network's hidden layer in this step. The number of output neurons is set to equal to the number of elements in the target vector as 32 (is total of RPs). This is a trial and error, we change gradually the number of hidden neuron to find a good performing network. This number is increased from default of 10 to 300, and the number of hidden neuron of 300 gives the best performing network.

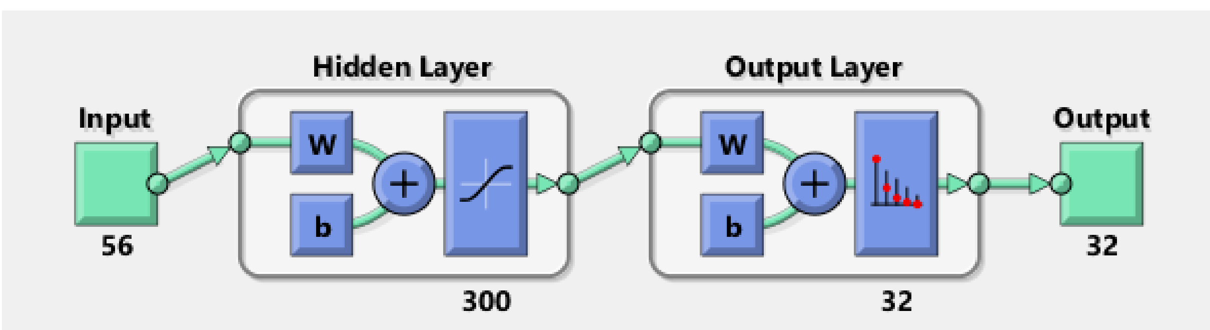


Figure 28: The network architecture in NN Pattern Recognition.

4.2.2.4 Train network

Train the network to classify the inputs according to the targets.

**** Plot the confusion.***

The confusion matrices of training, testing, validation, and the combination of three kinds are displayed as *Figure 29* below. This format is difficult to read due to inadequacy of

Matlab plotting of this matrices (we cannot zoom it). The rows shows the target class while the columns shows the output class. The green cells indicate the high numbers of correct responses while the red cells show the low numbers of incorrect responses.

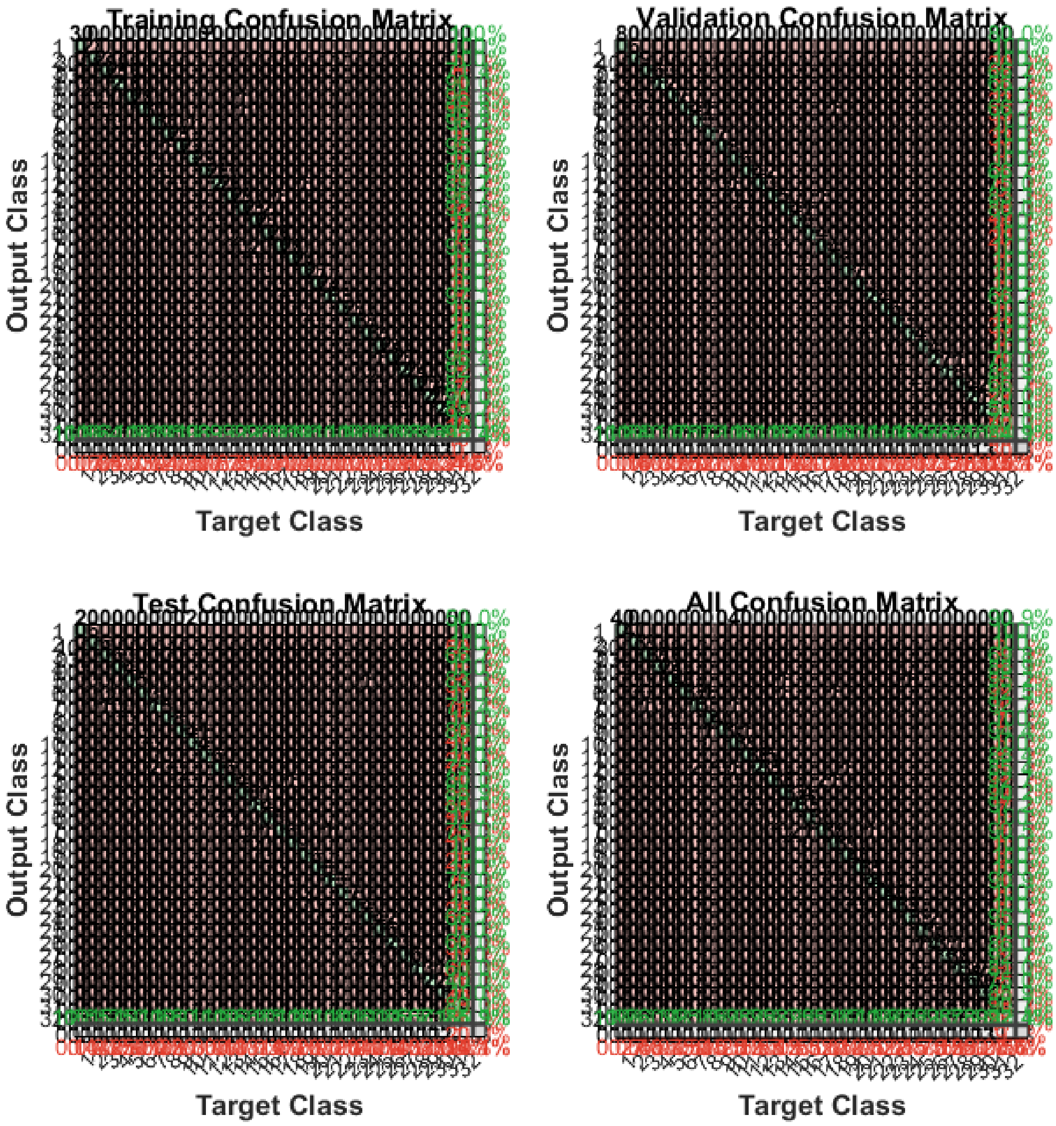


Figure 29: Plot the confusion in Pattern Recognition for Wi-Fi.

**** Plot the ROC curve.***

One of the important tools for assessing classification models is ROC curve. The plot of ROC curve have a axis as the true positive rate (TPR), another is the false positive rate (FPR). Each class are illustrated by each colored line. The curve with the spots in the upper-left corner, having 100% TPR and FPR shows the perfect test. The area under the curve (AUC) is a measure of how well the classifier is. AUC of 1 presents a perfect classifier. The ROC curve of all kinds are shown as figure below. With AUC values near to the 1, the graph shows that the model has a good measure of dissociation.

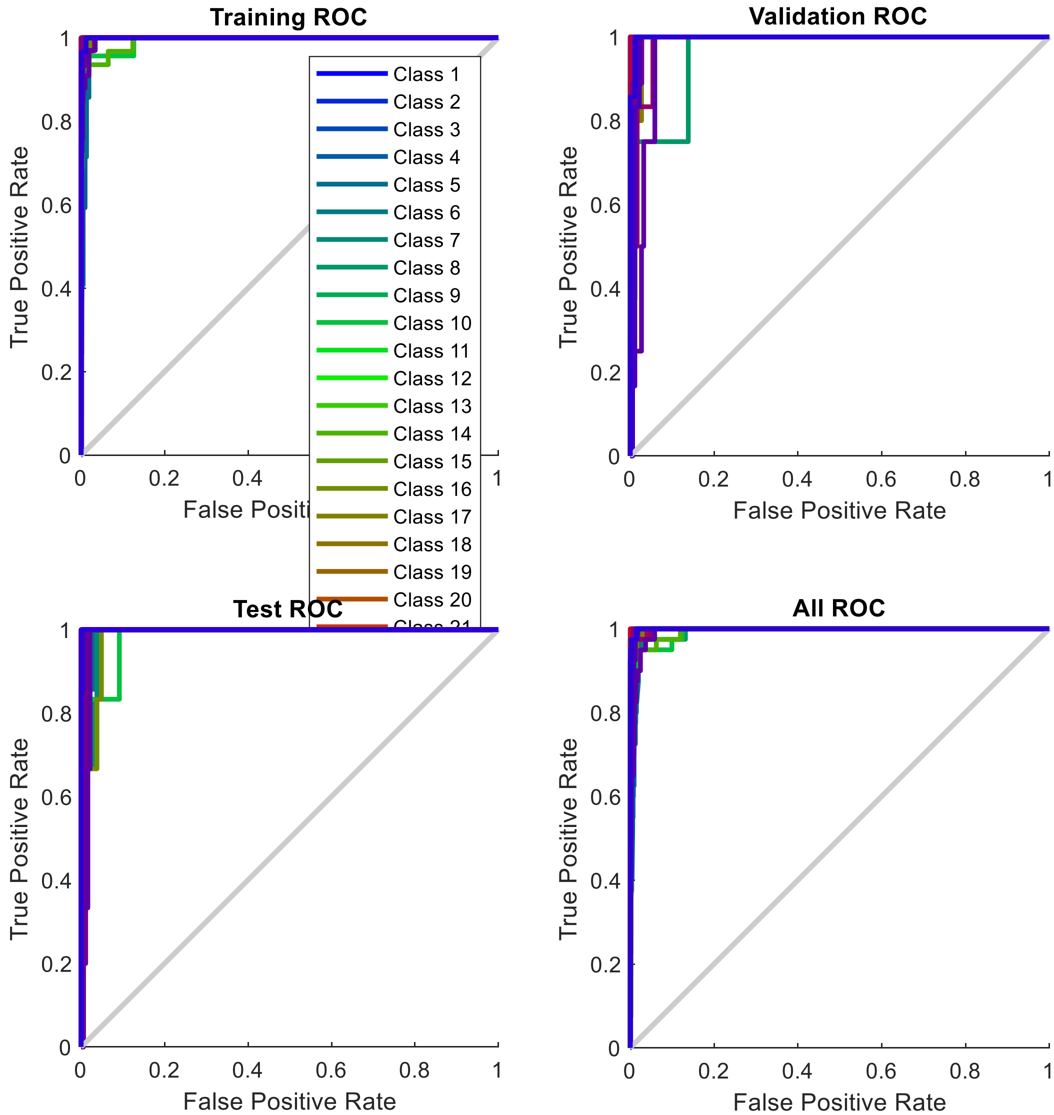


Figure 30: Plot the ROC curve in Pattern Recognition for Wi-Fi.

**** Plot the performance.***

Figure 31 presents the cross-entropy of Pattern Recognition model. According to the graph, the least cross-entropy in the validation phase is at epoch 75 which is the best validation performance, with value is 0.01181. The training model carry on in 5 error repetitions and stop at epoch 80.

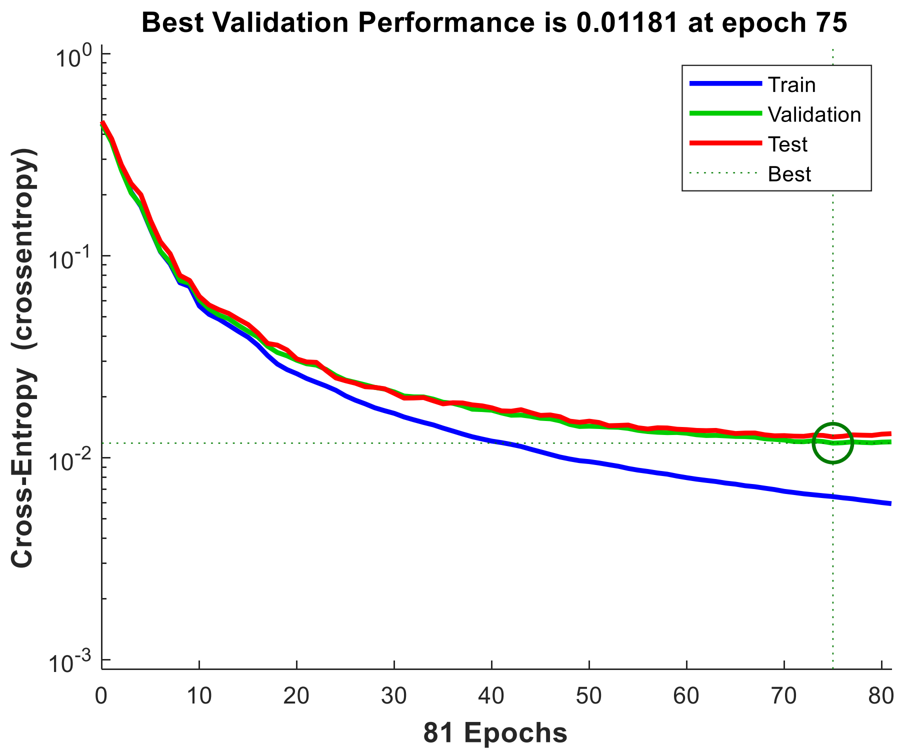


Figure 31: Plot the performance in Pattern Recognition for Wi-Fi.

**** Plot the error histogram.***

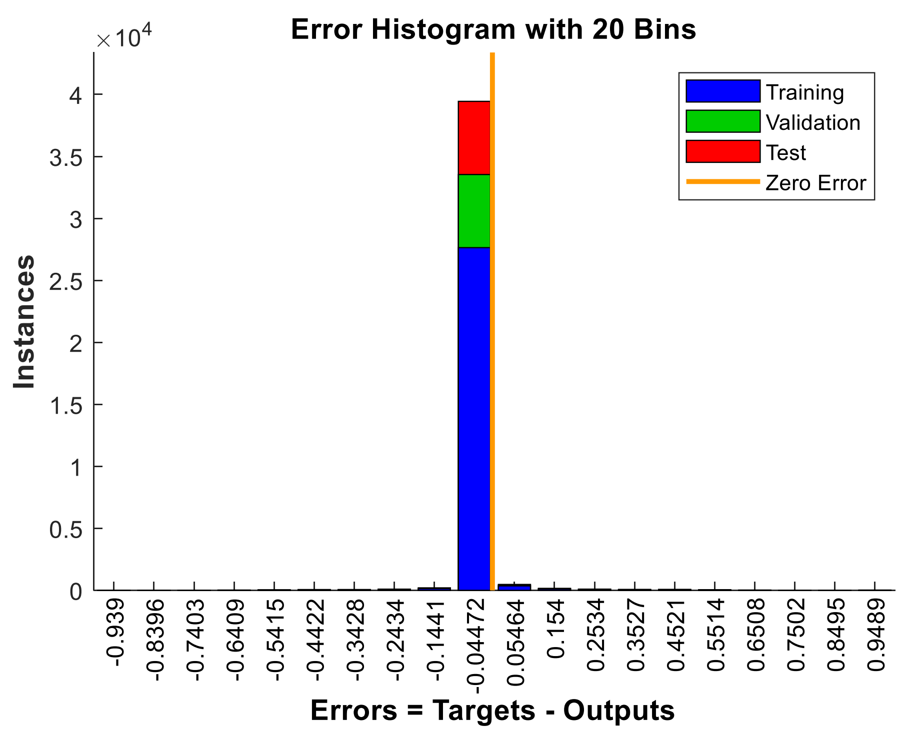


Figure 32: Plot the error histogram in Pattern Recognition for Wi-Fi.

The Error Histogram illustrates the dissimilarity between the targets and the actual outputs. This error is reported after training process completed. The error histogram of the trained neural network for the training, validation and testing kinds are showed in the *Figure 32*.

Bins are the number of vertical bars on the diagram. Range of the total error from -0.939 (leftmost bin) to 0.9489 (rightmost bin) is divided into 20 bins, so width of each bin is (0.9489 – (-0.939))/20 = 0.094395.

Each vertical bar shows the number of instances from the datasheet, which lies in a particular bin. At the mid of the above diagram, we have a bin corresponding to the error of -0.04472 and the height of that bin for training dataset (blue bars) is 2.75×10^4. It means that 2.75×10^4 instances from the training dataset have an error lies in the following range: (-0.04472 - 0.094395/2 , -0.04472 + 0.094395/2) or (-0.0919175, 0.0024775).

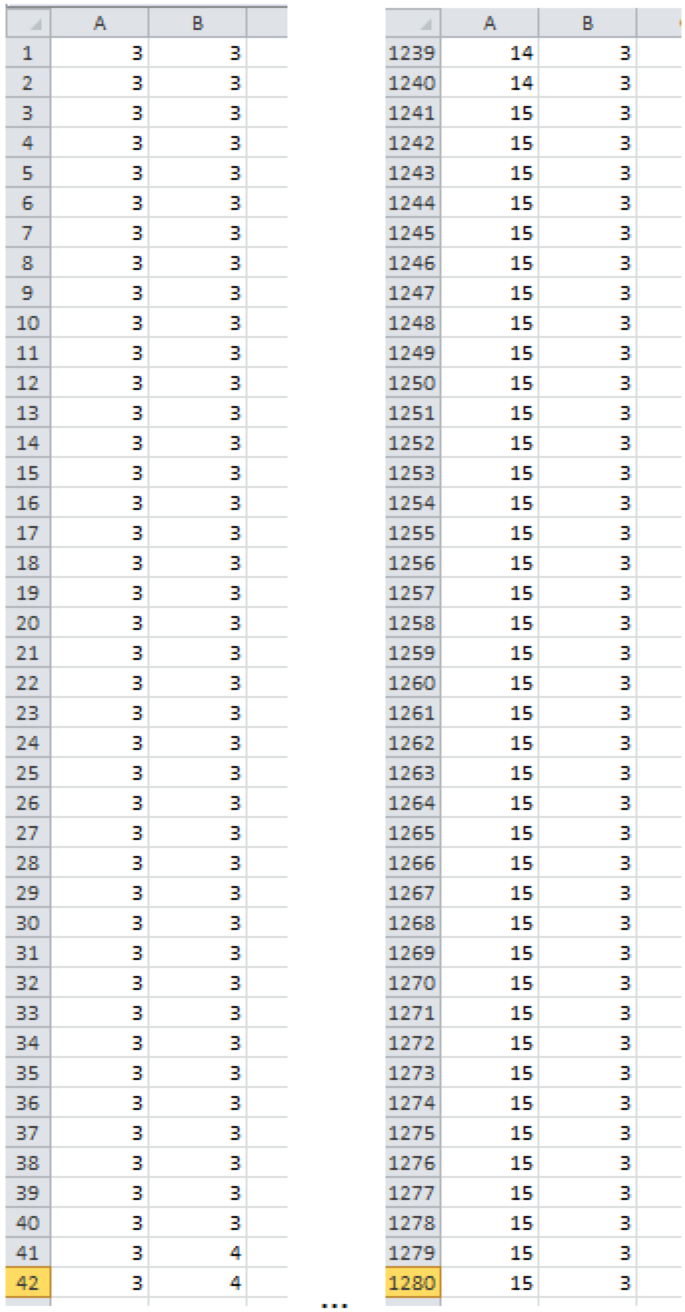
4.2.3 Neural Net Fitting.

The Neural Net (NN) Fitting app also uses a two-layer feed-forward network to solve a data-fitting problem. In fitting problems, we want a neural network to map between a data set of numeric inputs and a set of numeric targets. The NN Fitting app helps us select data, create and train a network and evaluate its performance using mean squared error and regression analysis.

4.2.3.1 Preparation to train.

*** Select data.**

The NN Fitting needs Input data and the Target data to train. The input data is similar to the input data used in the NN Pattern Recognition, a 1280x56 matrix. The target data is a 1280x2 matrix containing XY coordinate position of all RPs. Each RP have 40 rows of this corresponding to 40 scans, therefore there are total 1280 rows.



	A	B
1	3	3
2	3	3
3	3	3
4	3	3
5	3	3
6	3	3
7	3	3
8	3	3
9	3	3
10	3	3
11	3	3
12	3	3
13	3	3
14	3	3
15	3	3
16	3	3
17	3	3
18	3	3
19	3	3
20	3	3
21	3	3
22	3	3
23	3	3
24	3	3
25	3	3
26	3	3
27	3	3
28	3	3
29	3	3
30	3	3
31	3	3
32	3	3
33	3	3
34	3	3
35	3	3
36	3	3
37	3	3
38	3	3
39	3	3
40	3	3
41	3	4
42	3	4

...

	A	B
1239	14	3
1240	14	3
1241	15	3
1242	15	3
1243	15	3
1244	15	3
1245	15	3
1246	15	3
1247	15	3
1248	15	3
1249	15	3
1250	15	3
1251	15	3
1252	15	3
1253	15	3
1254	15	3
1255	15	3
1256	15	3
1257	15	3
1258	15	3
1259	15	3
1260	15	3
1261	15	3
1262	15	3
1263	15	3
1264	15	3
1265	15	3
1266	15	3
1267	15	3
1268	15	3
1269	15	3
1270	15	3
1271	15	3
1272	15	3
1273	15	3
1274	15	3
1275	15	3
1276	15	3
1277	15	3
1278	15	3
1279	15	3
1280	15	3

Figure 33: The format of target data in NN Fitting.

*** Validation and test data.**

The original data are randomly divided into three kinds similar to the Neural Net Pattern Recognition case above, with Training: 70%, Validation: 15% and Testing: 15%.

*** Set the network architecture.**

Network architecture have the number of output neurons as 2, equal to the number of elements in the target vector. Similar to setting the network architecture in Neural Net Pattern Recognition, the number of hidden neuron is set as 300. From the default value is 10, we increase gradually to find the best performing network. Through trial and error, 300 gives the best network performance.

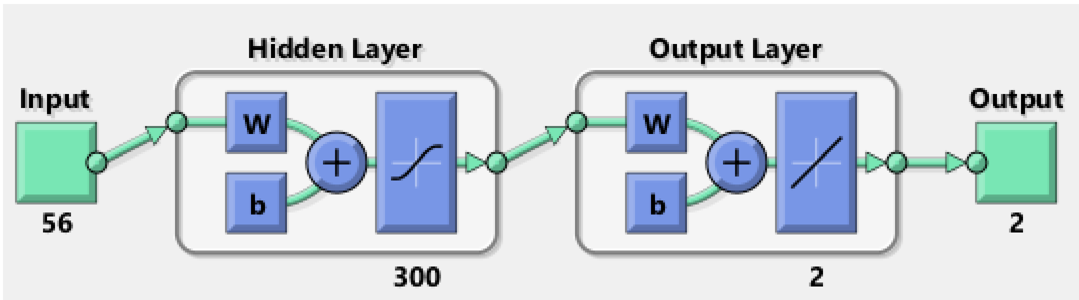


Figure 34: The network architecture in NN Fitting.

4.2.3.2 Train network

Train the network to fit the inputs and the targets. Levenberg-Marquardt is the algorithm chosen to train. Although this algorithm needs more memory, it requires less time than others. Training automatically stops when generalization stops improving, as indicated by an increase in the mean square error of the validation samples. When the validation error does not decline in six iterations, the training stops. The error rate after training is displayed below.

Results

	Samples	MSE	R
Training:	896	2.27625e-2	9.98591e-1
Validation:	192	1.32159e-0	9.30552e-1
Testing:	192	1.49842e-0	9.15129e-1

Figure 35: Result of error rate in Fitting for Wi-Fi.

**** Plot the regression.***

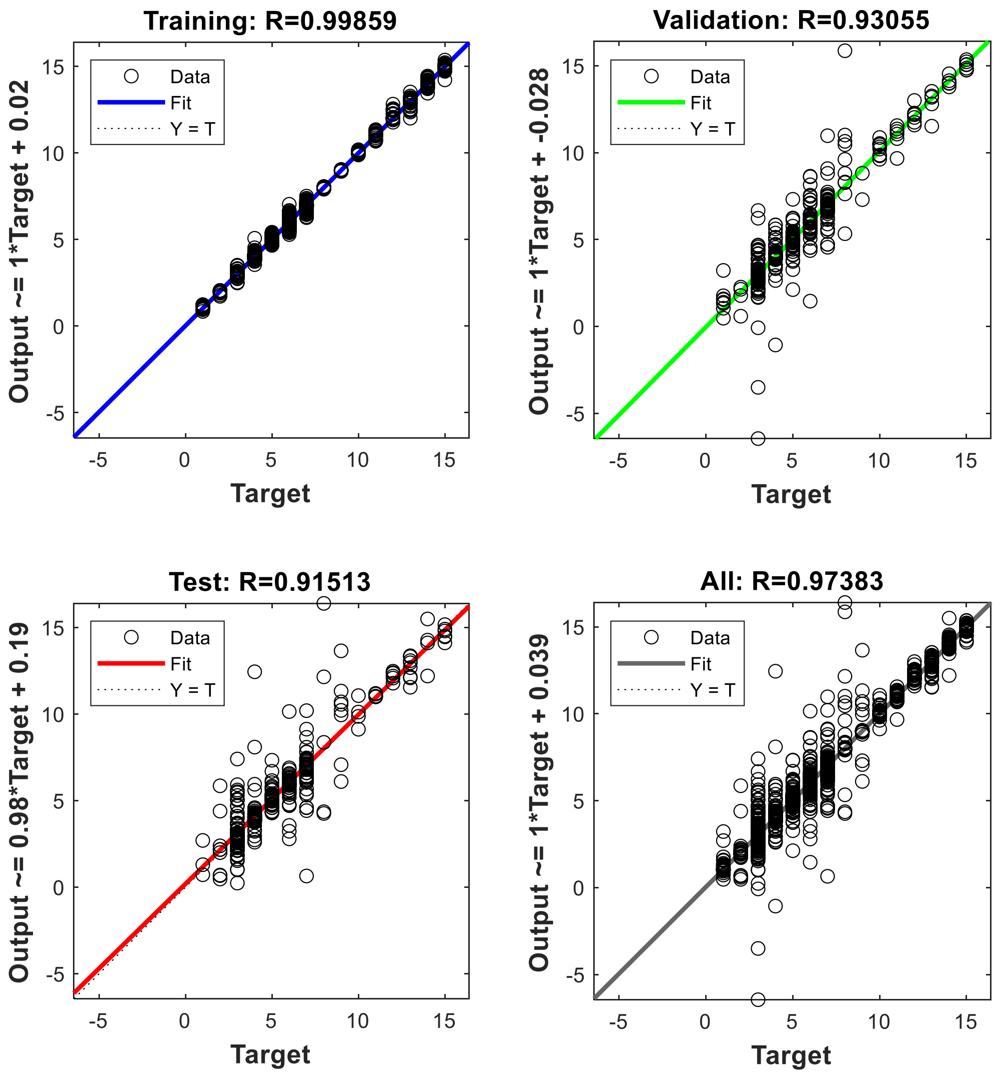


Figure 36: Plot the regression in Fitting for Wi-Fi.

The correlation between the target and output data of all stages (training, validation, test and combined all) are presented in *Figure 36*, via desirable correlation coefficient (R values). The 45 degree line in the graph indicates the network outputs equal to the targets. The fit is perfect when the data fall along this line. For the training data, this is almost the case, while for data not used in training, this will be worse.

**** Plot the performance.***

Mean Squared Error for train, validation, and test stages of Fitting model are displayed in *Figure 37*. According to the graph, the lowest MSE in the validation stage is at epoch 4 which is the best validation performance, with value is 1.3216. The training model carries on in 6 error repetitions and stop at epoch 10.

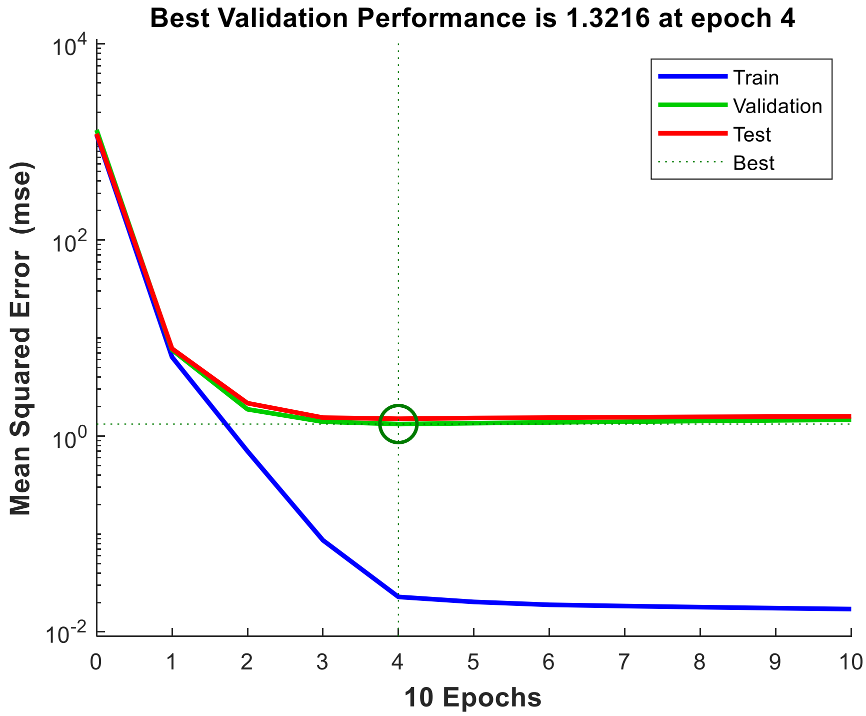


Figure 37: Plot the performance in Fitting for Wi-Fi.

**** Plot the error histogram.***

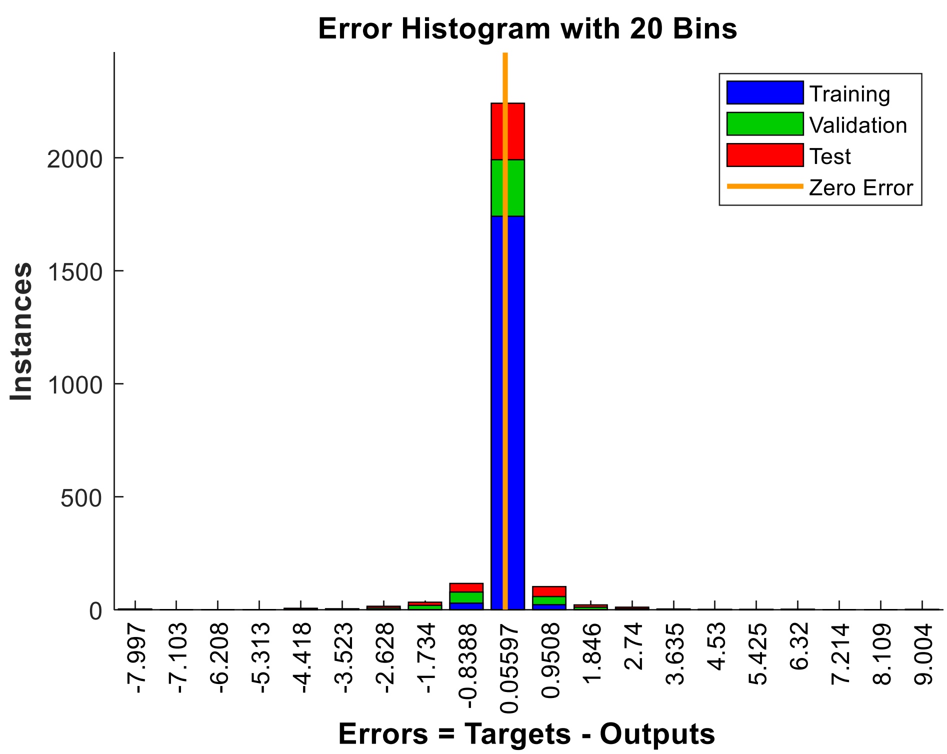


Figure 38: Plot the error histogram in Fitting for Wi-Fi.

Range of the total error from -7.997 (leftmost bin) to 9.004 (rightmost bin) is divided into 20 bins, so width of each bin is (9.004 – (-7.997))/20 = 0.85005.

At the mid of the above graph, we have a bin corresponding to the error of 0.05597. It means that 2250 instances from the training, validation and test dataset have an error lies in the following range: (0.05597 - 0.85005/2 , 0.05597 + 0.85005/2) or (-0.369055, 0.480995).

The above figures show that the error histogram for the training, validation and test kinds are delivered around zero error in a fairly good scope. The errors of most instances in training, validation and test dataset fall between -0.369005 and 0.480995, corresponding to from -1.1m to 1.4m in grid distances (3m). The dissimilarity between the targets and the outputs is less than 1.5m, this is good performance.

4.3 Training of the machine learning model for combined Wi-Fi and BLE.

The training data in this experiment are combined from the Wi-Fi fingerprinting data above and the BLE fingerprinting data. There are four locations for BLE equipment in the experimental area to generate signals to the RPs. The parameters are set for training in this case as in the case of Wi-Fi. The results obtained after the training with three machine learning applications are shown below.

4.3.1 Classification Learner.

Table 6 below shows the accuracy percentage in model for combined Wi-Fi and BLE. The model type having the best accuracy score is the Bagged Trees, with percentage of 92.1%.

No.	Model Type	Accuracy
Model 1.1	Fine Tree	88.1%
Model 1.2	Medium Tree	45.2%
Model 1.3	Coarse Tree	15.2%
Model 1.7	Kernel Naive Bayes	85.5%
Model 1.8	Linear SVM	90.3%
Model 1.9	Quadratic SVM	91.0%
Model 1.10	Cubic SVM	90.9%

Model 1.11	Fine Gaussian SVM	62.4%
Model 1.12	Medium Gaussian SVM	88.4%
Model 1.13	Coarse Gaussian SVM	83.4%
Model 1.14	Fine KNN	83.5%
Model 1.15	Medium KNN	80.6%
Model 1.16	Coarse KNN	56.6%
Model 1.17	Cosine KNN	79.8%
Model 1.18	Cubic KNN	75.7%
Model 1.19	Weighted KNN	84.5%
Model 1.20	Boosted Trees	89.3%
Model 1.21	***Bagged Trees***	***92.1%***
Model 1.22	Subspace Discriminant	79.8%
Model 1.23	Subspace KNN	91.4%
Model 1.24	RUSBoosted Trees	45.2%

Table 6: The accuracy percentage in model for combined Wi-Fi and BLE.

The *Figure 39* bellow shows the confusion matrix with the number of observations of model type Bagged Trees. The blue diagonal cells illustrate that the class is classified correctly.

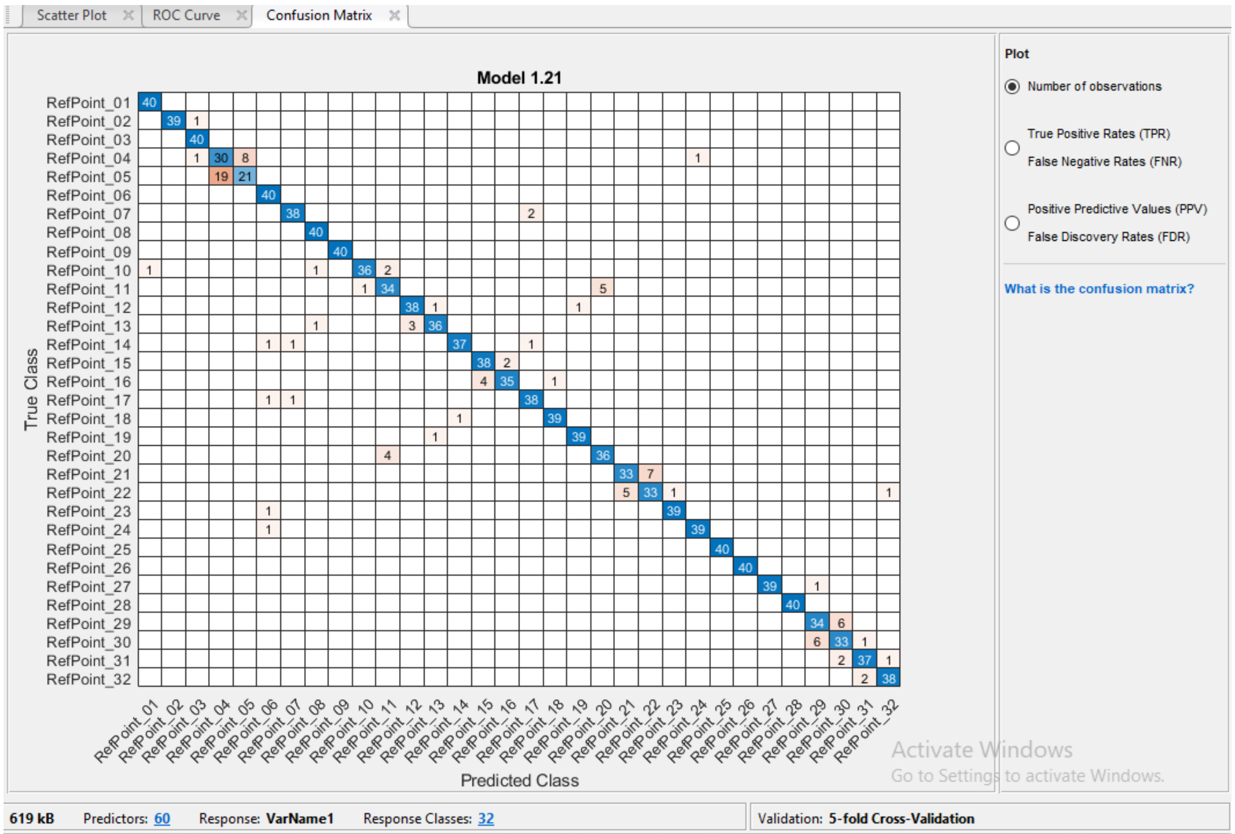


Figure 39: Confusion Matrix in model for combined Wi-Fi and BLE.

4.3.2 Neural Net Pattern Recognition.

The random division of samples into the training, validation and testing groups, and the error rate after training is displayed below.

Results	Samples	CE	%E
Training:	896	4.84340e-0	4.24107e-0
Validation:	192	14.16507e-0	10.93750e-0
Testing:	192	14.09368e-0	13.54166e-0

Figure 40: Result of error rate in Pattern Recognition for combined Wi-Fi and BLE.

**** Plot the ROC curve.***

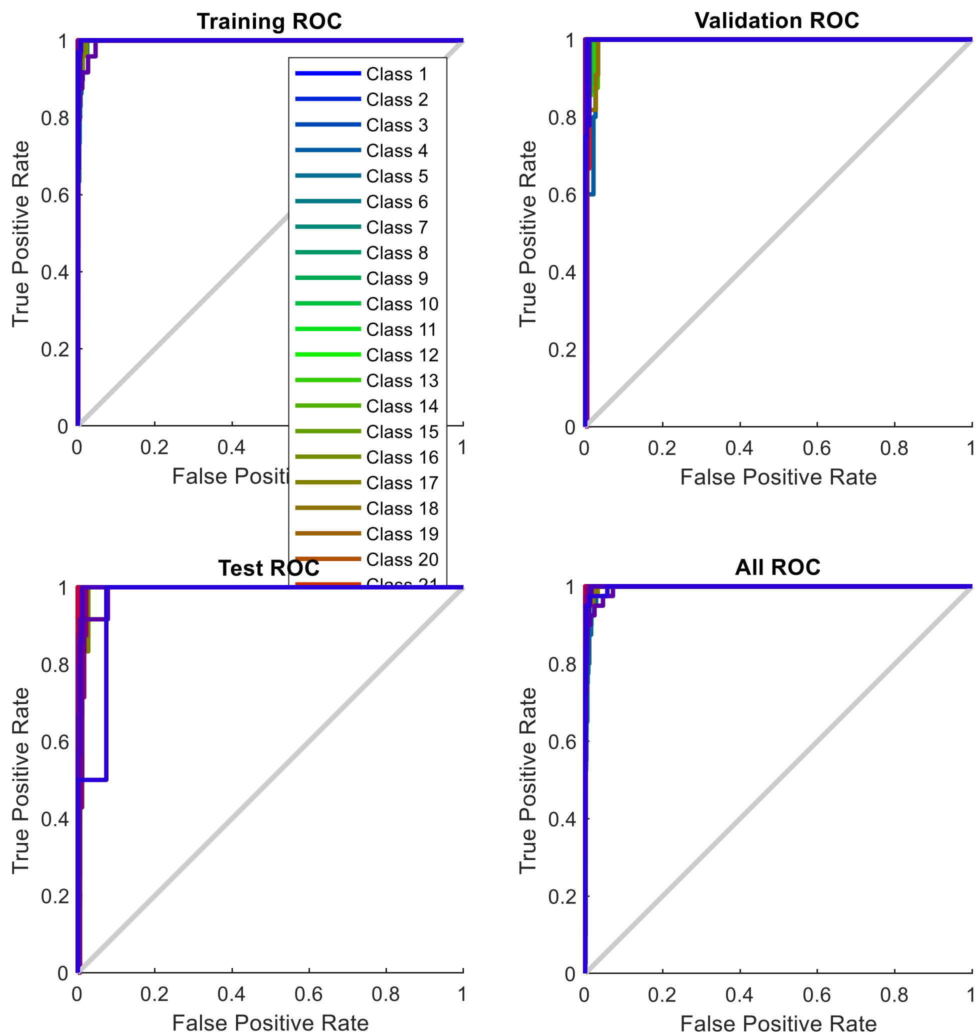


Figure 41: Plot the ROC curve in Pattern Recognition for combined Wi-Fi and BLE.

The ROC curve of training, validation, test and combined all are shown in *Figure 41*. With AUC values near to the 1, the graph shows that the model has a good measure of dissociation.

**** Plot the performance.***

Figure 42 shows the cross-entropy of Pattern Recognition model for combined Wi-Fi and BLE. According to the graph, the least cross-entropy in the validation phase is appeared at epoch 72 which is the best validation performance, with value is 0.010191. The training model carries on in 4 error repetitions and stop at epoch 78.

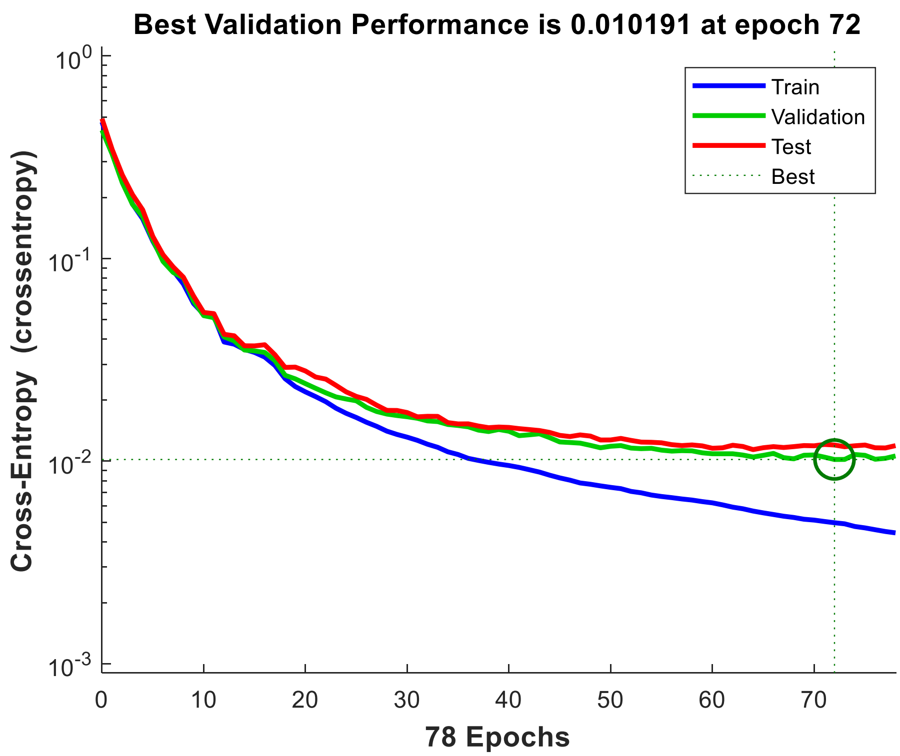


Figure 42: Plot the performance in Pattern Recognition for combined Wi-Fi and BLE.

**** Plot the error histogram.***

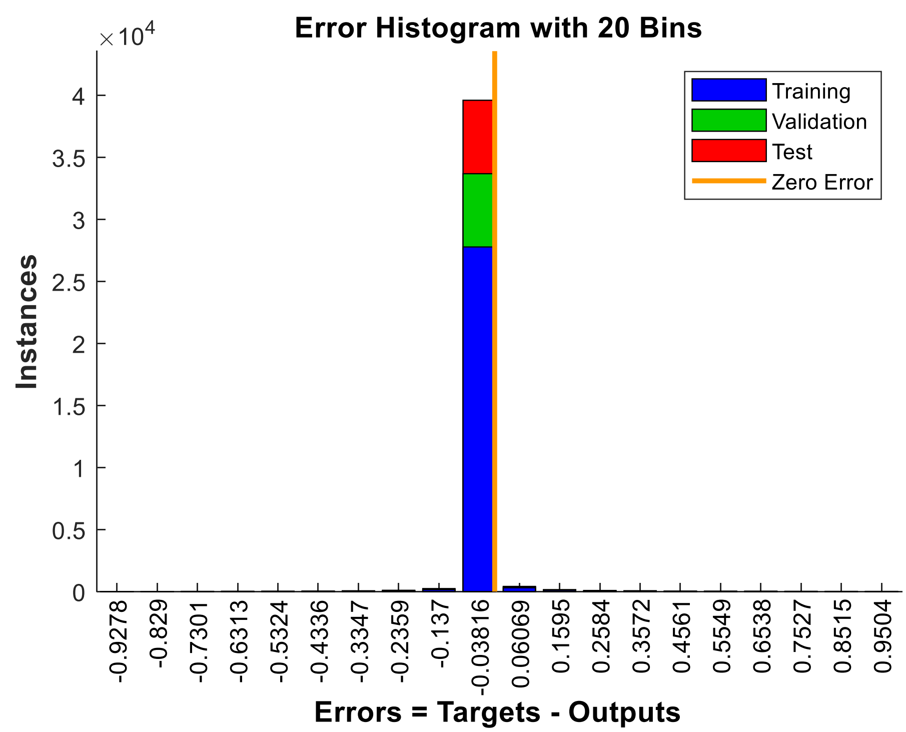


Figure 43: Plot error histogram in Pattern Recognition for combined Wi-Fi and BLE.

Range of the total error from -0.9278 (leftmost bin) to 0.9504 (rightmost bin) is divided into 20 bins, so width of each bin is (0.9504 – (-0.9278))/20 = 0.09391.

At the mid of the above graph, we have a bin corresponding to the error of -0.03816. It means that 4x10^4 instances from the training, validation and test dataset have an error lies in the following range: (-0.03816 - 0.09391/2, -0.03816 + 0.09391/2) or (-0.085115, 0.008795).

The *Figure 43* shows that the error histogram for the training, validation and test kinds are delivered around zero error in a fairly good scope. The error of around 4x10^4 instances in training, validation and test dataset stayed in the range from -0.085115 to 0.008795.

4.3.3 Neural Net Fitting.

The random division of samples into the training, validation and testing groups, and the error rate after training is displayed below.

Results

	Samples	MSE	R
Training:	896	1.95272e-1	9.89458e-1
Validation:	192	3.42945e-0	8.34045e-1
Testing:	192	3.41448e-0	8.17213e-1

Figure 44: Result of error rate in Fitting for combined Wi-Fi and BLE.

**** Plot the performance.***

Mean Squared Error for train, validation, and test stages of Fitting model (combined Wi-Fi and BLE) are displayed in *Figure 45*. According to the graph, the lowest MSE in the validation stage is appeared at epoch 6 which is the best validation performance, with value is 3.4295. The training model carries on in 6 error repetitions and stop at epoch 12.

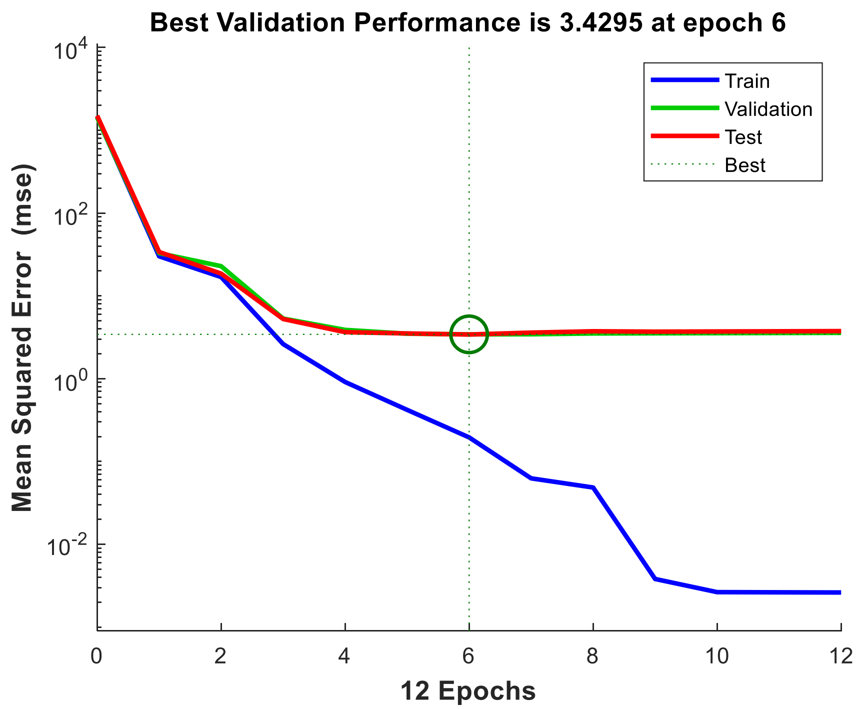


Figure 45: Plot the performance in Fitting for combined Wi-Fi and BLE.

**** Plot the error histogram.***

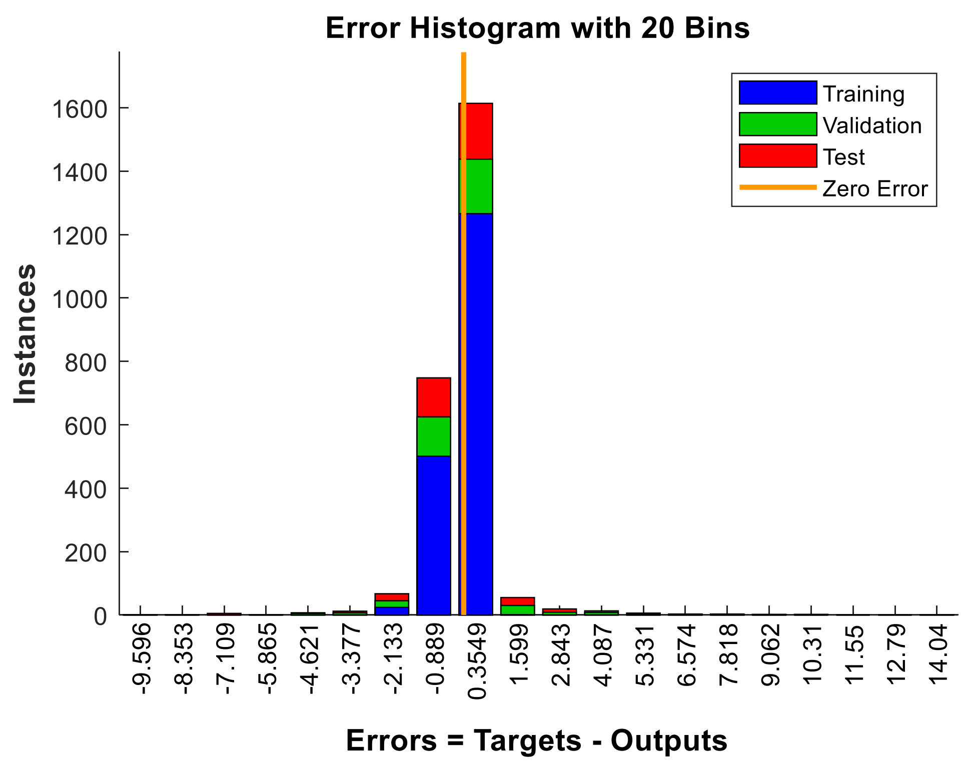


Figure 46: Plot the error histogram in Fitting for combined Wi-Fi and BLE.

Range of the total error from -9.596 (leftmost bin) to 14.04 (rightmost bin) is divided into 20 bins, so width of each bin is (14.04 – (-9.596))/20 = 1.1818.

At the mid of the above graph, we have a bin corresponding to the error of 0.3549. It means that 1600 instances from the training, validation and test dataset have an error lies in the following range: (0.3549 - 1.1818/2 , 0.3549 + 1.1818/2) or (-0.236, 0.9458).

The above figures show that the error histogram for the training, validation and test kinds are delivered around zero error in a fairly good scope. The errors of most instances in training, validation and test dataset fall between -0.236 and 0.9458, corresponding to from -0.7m to 2.8m in grid distances (3m).

**** Plot the regression.***

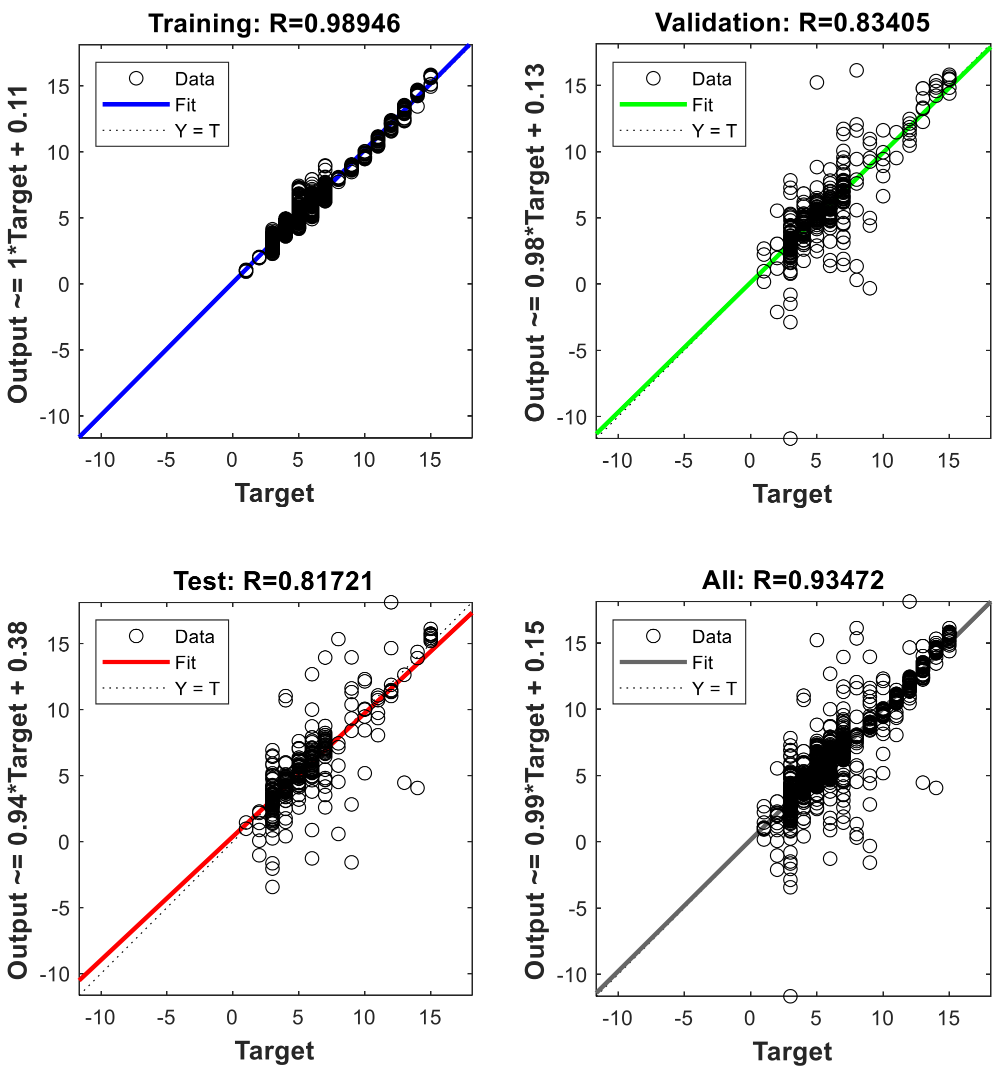


Figure 47: Plot the regression in Fitting for combined Wi-Fi and BLE.

The correlation between the target and output data for training, validation, test and combined all are presented in *Figure 47*, via desirable correlation coefficient (R values).

4.4 Evaluate and compare the system's performance

4.4.1 Performance evaluation of machine learning techniques for Wi-Fi fingerprint data.

4.4.1.1 Classification Learner.

According to *Table 5* in 4.2.1.1, the obtained accuracy of machine learning techniques via Classification Learner tool ranges from 15.2% to 91.7% with different classifiers. The Bagged Trees classifier obtains the highest accuracy of 91.7%. On average, among the machine learning techniques, the Naive Bayes and SVM techniques have the best average accuracy rate, 84.3% and 83.4% respectively.

The confusion matrix plot and ROC curve help to examine performance per class of a training model. As seen in *Figure 23* in 4.2.1.2, the matching value at the diagonal cells of row and column RefPoint_04, 05 are the worst, corresponding to the lowest number of APs detected at these locations (as *Table 7* below). This shows that the number of APs detected at RPs affects on the accuracy of training model. Performance at RPs that have more APs are better than at RPs with less detected APs.

Location	Maximum number of APs detected	Minimum number of APs detected	Average number of APs detected	Matching value in model type Bagged Trees (/40)
RefPoint_01	25	15	21	40
RefPoint_02	21	17	20	40
RefPoint_03	19	5	12	39
RefPoint_04	15	5	6	33
RefPoint_05	11	5	6	20
RefPoint_06	15	9	10	39
RefPoint_07	20	14	16	40
RefPoint_08	20	10	14	40
RefPoint_09	21	19	20	40
RefPoint_10	22	10	16	36
RefPoint_11	21	15	16	38
RefPoint_12	21	16	19	38
RefPoint_13	21	19	21	35
RefPoint_14	17	10	16	38
RefPoint_15	21	16	20	36

RefPoint_16	22	16	19	31
RefPoint_17	17	14	17	40
RefPoint_18	22	13	16	38
RefPoint_19	24	19	21	39
RefPoint_20	30	15	19	39
RefPoint_21	26	19	22	33
RefPoint_22	27	14	21	32
RefPoint_23	35	21	28	39
RefPoint_24	22	15	20	39
RefPoint_25	34	15	23	39
RefPoint_26	33	16	23	40
RefPoint_27	24	9	16	37
RefPoint_28	18	13	16	39
RefPoint_29	21	10	17	31
RefPoint_30	21	15	16	31
RefPoint_31	20	15	16	36
RefPoint_32	18	11	16	39

Table 7: The number of APs scanned at RPs and the matching value.

4.4.1.2 NN Pattern Recognition and Fitting.

Table 8 below shows the training results of the NN Pattern Recognition and Fitting for Wi-Fi fingerprint data.

Types	**Samples**	**NN Pattern Recognition**		**NN Fitting**	
		CE	%E	MSE	R
Training	896	5.31001e-0	4.57589e-0	2.27625e-2	9.98591e-1
Validation	192	15.00388e-0	15.10416e-0	1.32159e-0	9.30552e-1
Testing	192	15.00545e-0	14.06250e-0	1.49842e-0	9.15129e-1

Table 8: Training results of NN Pattern Recognition and Fitting.

The Pattern Recognition evaluates the results based on cross-entropy (CE) and percent misclassification error (%E). Lower CE value is a better classification and a value of 0 means no error. %E shows the fraction of samples which are misclassified. A value of 100 means maximum misclassifications while zero indicates no misclassifications. For test data, the %E is commonly less than 30% to ensure that the accuracy of classification is comparable to that we could achieve with the classifier models (more than 60%).

The NN Fitting evaluates the results based on Mean Squared Error (MSE) and Regression (R). MSE is the average squared difference between outputs and targets, the lower MSE values are the better system (this is in terms of grid distances). The correlation between outputs and targets is measured by R values. R = 1 represents a close relationship while R = 0 serves as a random relationship. The values achieved in *Table 8* show that the data fitting is quite accurate.

4.4.2 The performance of the machine learning model in methods.

4.4.2.1 Accuracy comparison of model for Wi-Fi, BLE, and combined Wi-Fi and BLE in Classification Learner.

No.	**Model Type**	**Wi-Fi**	**BLE**	**Combined**
Model 1.1	Fine Tree	88.9%	51.0%	88.1%
Model 1.2	Medium Tree	45.5%	43.5%	45.2%
Model 1.3	Coarse Tree	15.2%	14.8%	15.2%
Model 1.7	Kernel Naive Bayes	84.3%	47.2%	85.5%
Model 1.8	Linear SVM	89.6%	51.9%	90.3%
Model 1.9	Quadratic SVM	89.5%	**55.0%**	91.0%
Model 1.10	Cubic SVM	88.9%	53.0%	90.9%
Model 1.11	Fine Gaussian SVM	66.2%	51.5%	62.4%
Model 1.12	Medium Gaussian SVM	87.0%	53.4%	88.4%
Model 1.13	Coarse Gaussian SVM	79.4%	48.3%	83.4%
Model 1.14	Fine KNN	81.2%	49.7%	83.5%
Model 1.15	Medium KNN	82.0%	52.3%	80.6%
Model 1.16	Coarse KNN	48.1%	38.8%	56.6%
Model 1.17	Cosine KNN	81.6%	49.3%	79.8%
Model 1.18	Cubic KNN	79.1%	52.7%	75.7%
Model 1.19	Weighted KNN	84.2%	52.4%	84.5%
Model 1.20	Boosted Trees	87.0%	46.6%	89.3%
Model 1.21	Bagged Trees	***91.7%***	49.9%	**92.1%**
Model 1.22	Subspace Discriminant	75.3%	46.0%	79.8%
Model 1.23	Subspace KNN	88.6%	30.9%	91.4%
Model 1.24	RUSBoosted Trees	45.6%	43.5%	45.2%

Table 9: Accuracy of model for Wi-Fi, BLE, and combined Wi-Fi and BLE .

The similar experiment is deployed in the same area for BLE fingerprinting provide the results in BLE column of above *Table 9*. There are only four BLE beacons in the experimental area to generate signals to the RPs, therefore the accuracy percentage in BLE fingerprinting is low. While the combined Wi-Fi and BLE fingerprinting data gives slightly better results than Wi-Fi fingerprinting data.

4.4.2.2 The performance comparison between Wi-Fi and combined model.

**** NN Pattern Recognition.***

Table 10 below shows the training results of the NN Pattern Recognition for Wi-Fi and combined Wi-Fi and BLE fingerprint data.

Types	**Samples**	**Cross-Entropy (CE)**		**Misclassification error (%E)**	
		Wi-Fi	***Combined***	***Wi-Fi***	***Combined***
Training	896	5.31001e-0	4.8434e-0	4.57589e-0	4.24107e-0
Validation	192	15.00388e-0	14.16507e-0	15.10416e-0	10.93750e-0
Testing	192	15.00545e-0	14.09368e-0	14.06250e-0	13.54166e-0

Table 10: Training results of Recognition for Wi-Fi & combined Wi-Fi and BLE.

The performance of the model is evaluated via the validation dataset. In NN Pattern Recognition, the best validation performance is 0.01181 at epoch 75 (*Figure 31*) for Wi-Fi model, while the value is 0.010191 at epoch 72 (*Figure 42*) for the combined Wi-Fi and BLE model.

The difference between the target value and predicted values after training are displayed in the error histogram graph. In NN Pattern Recognition, both case (with and without BLE), The error range of around $4x10^4$ instances of the training, validation and test falls in the bin around zero value. For Wi-Fi model, the range is from -0.0919175 to 0.0024775 (*Figure 32*) while combined Wi-Fi and BLE model even has the better range from -0.085115 to 0.008795 (*Figure 43*). This means, the output data and target value is close, and NN model is good.

**** NN Fitting.***

Table 11 below shows the training results of the NN Fitting for Wi-Fi and combined Wi-Fi and BLE fingerprint data.

Types	Samples	Mean Squared Error (MSE)		Regression (R)	
		Wi-Fi	*Combined*	*Wi-Fi*	*Combined*
Training	896	2.27625e-2	1.95272e-1	9.98591e-1	9.89458e-1
Validation	192	1.32159e-0	3.42945e-0	9.30552e-1	8.34045e-1
Testing	192	1.49842e-0	3.41448e-0	9.15129e-1	8.17213e-1

Table 11: Training results of NN Fitting for Wi-Fi and combined Wi-Fi and BLE.

In NN Fitting, the best validation performance for Wi-Fi model is 1.3216 at epoch 4 (of 10 epochs), is showed in *Figure 37*. With combined Wi-Fi and BLE case, the best validation performance is 3.4295 at epoch 6 (of 12 epochs), is displayed in *Figure 45*.

The error histogram graphs show that the errors of most instances fall in range from -0.369005 to 0.480995, corresponding to from -1.1m to 1.4m in grid distances, in Wi-Fi model (*Figure 38*) while the error range lies on between -0.236 and 0.9458, corresponding to from -0.7m to 2.8m in grid distances, in combined Wi-Fi and BLE model (*Figure 46*).

***** In this experiment, the combined Wi-Fi and BLE model gives better validation performance than the Wi-Fi model in the NN Pattern Recognition. Meanwhile, the performance of the Wi-Fi model is better than combined model in NN Fitting. This may not be a definitive result. Probably it is caused by the differences in running the NN model.

4.5 Neural Net Fitting testing on Wi-Fi fingerprinting data.

This step is an independent test. New test fingerprints are collected and processed by the already trained network to find the actual performance that could be achieved with the method in the online phase.

4.5.1 Prepare the test input data.

The implementation steps in this stage are similar to data collection stage for training. There are eight test-points selected in experiment area with XY coordinates like *Table 12* below.

Test-points	***X***	***Y***
TestingPoint_01	3	5.5
TestingPoint_02	4.5	6.5
TestingPoint_03	5.5	5.5
TestingPoint_04	4.5	4.5
TestingPoint_05	5.5	3
TestingPoint_06	7	1.5
TestingPoint_07	9.5	3
TestingPoint_08	12.5	3

Table 12: Coordinates of test-points.

	A	B	C	D	E	F	G
1		58:AC:78:89:EE:E0	58:AC:78:89:EE:E4	58:AC:78:89:EE:E2	58:AC:78:89:EE:E1	58:AC:78:89:EE:E3	58:AC:78:9C:B7:F0
2	TestingPoint_01	-48	-48	-48	-48	-48	-110
3	TestingPoint_01	-42	-42	-42	-42	-42	-110
4	TestingPoint_01	-44	-44	-44	-44	-44	-86
5	TestingPoint_01	-40	-41	-41	-41	-41	-110
6	TestingPoint_01	-43	-43	-43	-43	-43	-110
7	TestingPoint_01	-40	-41	-41	-41	-41	-83
8	TestingPoint_01	-42	-42	-42	-42	-42	-110
9	TestingPoint_01	-43	-42	-43	-41	-43	-82
10	TestingPoint_01	-45	-45	-45	-45	-45	-110
11	TestingPoint_01	-45	-45	-45	-45	-45	-110
12	TestingPoint_02	-64	-64	-64	-64	-64	-110
13	TestingPoint_02	-65	-65	-65	-65	-64	-110
14	TestingPoint_02	-63	-63	-63	-63	-63	-84
15	TestingPoint_02	-67	-67	-67	-67	-67	-110
16	TestingPoint_02	-64	-64	-65	-64	-64	-110
17	TestingPoint_02	-65	-65	-65	-65	-65	-110
18	TestingPoint_02	-65	-65	-65	-65	-65	-110
19	TestingPoint_02	-64	-65	-65	-65	-65	-110
20	TestingPoint_02	-64	-64	-64	-65	-64	-110
21	TestingPoint_02	-64	-64	-64	-65	-65	-110
22	TestingPoint_03	-60	-60	-60	-60	-60	-79

Figure 48: The format of testing data file.

Taking 10 scans in ESP32 board for each test-points to get the test input data. The input data are collected and converted to appropriate format like the training input data. *Figure 48* above shows the format of the test input data, this is an excel file of size 81 x 57. The first row presents all the MAC addresses scanned in testing phase, the list is the same with those in training phase. 80 followed rows (8 test-points and 10 scans per each test-point) are the RSSI values of APs received by the device at the test-points.

4.5.2 Testing Process.

Testing process is done in Matlab following these steps:

- Transpose the testing data to fit with the dimensions of the trained net.
- Test-points positions data is tailored like the net fitting targets file prepared for the reference points.
- Run the net with the transposed testing data to get the result. Then, transpose the result.
- Subtract (element-wise) the transposed result from the test-points positions data to get the X, Y errors.
- Convert the X, Y errors into distances by formula: Distances error= $\sqrt{X_{error}^2 + Y_{error}^2}$

The code used in Matlab as below:

```
load('Net_Fitting.mat')
testdata = transpose(data_2) %data_2: testing data.
result = net(testdata)
transpose_result = transpose(result)
errors = transpose_result - data_3  %data_3: testpoints position.
```

4.5.3 Testing Results.

The testing result for the trained network in the NN Fitting case is displayed below:

Test-points	***Average X error***	***Average Y error***	***Average error distance***
TestingPoint_01	-0.05971	-0.46823	0.47203
TestingPoint_02	-0.57133	0.04222	0.57289
TestingPoint_03	-0.70045	0.02782	0.70101
TestingPoint_04	-0.25065	0.27223	0.37005
TestingPoint_05	-1.15313	0.15249	1.16317
TestingPoint_06	-0.69539	0.78511	1.04879
TestingPoint_07	1.50201	-0.76896	1.68741
TestingPoint_08	1.71563	-0.02992	1.71589

Table 13: The testing result for the trained network.

According to the *Table 13*, the error distance of test-points falls within approximately 1.7 units from the real position. The minimum error distance is 0.37005 unit, equivalent to 1.110m (the experimental area is divided on the 3m x 3m grid, 0.37005 x 3m = 1.110m). The maximum error distance is 1.71589 units, equivalent to 5.148m. The average distance error of testing experiment is 0.99640 unit, equivalent to 2.899m. This means that the data obtained at experimental area used with the fitting NN model allows for localisation with reasonable accuracy, from 1m to 3m in most cases. The trained network has the ability to generalise and show coordinates of the test points that are different from the reference points.

CHAPTER 5: CONCLUSION AND FUTURE WORK

5.1 Conclusion.

The project focuses on the performance evaluation of Wi-Fi based indoor positioning systems. Besides, the performance evaluation between systems based on Wi-Fi, BLE and combined Wi-Fi and BLE also are carried out.

Collection process of fingerprinting data for ambient Wi-Fi and BLE was performed in a part of the first floor of the SCT building, where is covered by more than 50 APs. In addition, there are four BLE beacons in the experimental area. Training of the machine learning model for Wi-Fi fingerprinting data is implemented firstly, then performing with combined Wi-Fi and BLE model. The results obtained in these trainings, along with the training result of the machine learning model for BLE from the classmate's project, are used to compare and evaluate systems performance.

Performance evaluation of machine learning techniques for Wi-Fi fingerprint data shows that the number of APs detected at RPs affects on the accuracy of training model. Performance at RPs that have more APs are better than at RPs with less scanned APs. The results of the NN Pattern Recognition ensure that the accuracy of classification is comparable to that we could achieve with the classifier models. The values achieved in the NN Fitting describes the data fitting is quite accurate.

Accuracy comparison of model for Wi-Fi, BLE, and combined Wi-Fi and BLE in Classification Learner, the experimental results show that the accuracy percentage in BLE fingerprinting is the lowest - the cause may be due to suffering from insufficient number of beacons, while the combined Wi-Fi and BLE fingerprinting data gives a better result in a lots of model types. When comparing the performance between Wi-Fi and combined Wi-Fi and BLE model, the performance of the Wi-Fi model is better than combined model in NN Fitting, while the combined Wi-Fi and BLE model gives better validation performance than the Wi-Fi model in the NN Pattern Recognition.

The result obtained when performing testing step at experimental area used with the fitting NN model shows that the system accuracy in the test-points is positioning error less than 3.1m in 63% of cases. Comparing with the results in reference points, the error of most instances in test dataset falls between -1.1m and 1.4m. This can be explained as there might have been changes to the APs around the testing area. Due to the fact that the testing point fingerprints were collected some time after the original fingerprinting map.

With all above experimental results, we can conclude that the performance of Wi-Fi based indoor positioning system has been developed and evaluated with good results.

5.2 Future work.

The next step of this project is an attempt at the development of a demonstrator system to track the location of the device. In addition, it is necessary to design automatic data collection tools to increase the system's practicality.

Although the experimental results in this study look good, additional studies are needed before the system can be widely applied into practice. This experiment is based on the received Wi-Fi RSSI values, which are unstable signal sources depending on the number of APs at different times, while number of beacons are limited, so the accuracy of system is not stable. Future work might be building a system that incorporates additional signal sources, such as magnetic field strength in the area, to improve the accuracy of the positioning system.

APPENDIX

Test-points	*X error*	*Y error*	*Distance error*
TestingPoint_01	0.04664	-1.29487	1.29571
TestingPoint_01	-0.24445	-0.03232	0.24658
TestingPoint_01	-0.43438	0.66936	0.79796
TestingPoint_01	0.42414	-1.03091	1.11475
TestingPoint_01	-0.20956	0.01905	0.21042
TestingPoint_01	-0.36241	-0.32101	0.48413
TestingPoint_01	-0.97356	-0.78428	1.25016
TestingPoint_01	4.22037	-2.40586	4.85795
TestingPoint_01	-0.90830	0.84931	1.24352
TestingPoint_01	-2.15561	-0.35080	2.18397
TestingPoint_02	-0.60893	-0.20781	0.64341
TestingPoint_02	-0.41466	0.30480	0.51463
TestingPoint_02	-0.80101	-0.25751	0.84139
TestingPoint_02	-0.45627	0.38365	0.59612
TestingPoint_02	-0.56767	-0.01360	0.56783
TestingPoint_02	-0.71480	0.41718	0.82764
TestingPoint_02	-0.32976	0.16753	0.36987
TestingPoint_02	-0.54865	-0.09233	0.55636
TestingPoint_02	-0.70015	-0.15451	0.71699
TestingPoint_02	-0.57140	-0.12522	0.58496
TestingPoint_03	-0.30261	-0.76042	0.81842
TestingPoint_03	-0.33531	0.34574	0.48163
TestingPoint_03	0.29156	-0.56558	0.63630
TestingPoint_03	-1.06827	-0.12221	1.07523
TestingPoint_03	-0.85006	-0.37216	0.92795
TestingPoint_03	-0.99243	-0.01557	0.99255
TestingPoint_03	0.06920	-0.39426	0.40029
TestingPoint_03	-1.49745	1.05643	1.83259
TestingPoint_03	-2.07902	1.11365	2.35850
TestingPoint_03	-0.24015	-0.00743	0.24026
TestingPoint_04	2.09342	0.58841	2.17454
TestingPoint_04	0.10494	-0.17721	0.20595
TestingPoint_04	-0.10441	0.13646	0.17182
TestingPoint_04	-0.47712	0.07742	0.48336
TestingPoint_04	-0.50160	0.28295	0.57590
TestingPoint_04	-0.51624	0.46065	0.69188
TestingPoint_04	0.44137	-0.68116	0.81166
TestingPoint_04	-0.09691	0.63749	0.64481
TestingPoint_04	-3.28629	1.12334	3.47298

TestingPoint_04	-0.16369	0.27393	0.31911
TestingPoint_05	-1.21231	0.21862	1.23187
TestingPoint_05	-2.16208	0.36308	2.19235
TestingPoint_05	-2.58778	0.60718	2.65806
TestingPoint_05	-1.27548	-0.29278	1.30865
TestingPoint_05	-2.66871	0.80409	2.78722
TestingPoint_05	-1.59457	0.85035	1.80714
TestingPoint_05	0.68579	-0.23164	0.72386
TestingPoint_05	0.14270	-0.63159	0.64751
TestingPoint_05	0.51030	-0.20616	0.55037
TestingPoint_05	-1.36918	0.04379	1.36988
TestingPoint_06	-0.15785	1.98720	1.99346
TestingPoint_06	-3.90944	0.76356	3.98331
TestingPoint_06	-3.51422	-1.60162	3.86199
TestingPoint_06	1.05831	1.72297	2.02204
TestingPoint_06	1.15157	1.74077	2.08720
TestingPoint_06	-3.87400	2.29283	4.50166
TestingPoint_06	1.58421	1.42742	2.13243
TestingPoint_06	2.87175	-2.59376	3.86970
TestingPoint_06	-0.88668	0.97588	1.31854
TestingPoint_06	-1.27754	1.13589	1.70950
TestingPoint_07	3.78015	1.68423	4.13837
TestingPoint_07	2.43832	0.18467	2.44530
TestingPoint_07	-0.25143	-0.43462	0.50211
TestingPoint_07	2.42843	-1.38795	2.79708
TestingPoint_07	2.53444	-1.29552	2.84636
TestingPoint_07	3.62632	-2.20865	4.24598
TestingPoint_07	-3.35999	-2.92269	4.45327
TestingPoint_07	2.50389	-0.72416	2.60651
TestingPoint_07	-0.27642	-0.14406	0.31171
TestingPoint_07	1.59641	-0.44087	1.65617
TestingPoint_08	1.80836	-0.02190	1.80849
TestingPoint_08	1.78329	-0.00151	1.78329
TestingPoint_08	1.57701	0.05959	1.57814
TestingPoint_08	1.54769	-0.07091	1.54932
TestingPoint_08	1.53075	-0.09013	1.53340
TestingPoint_08	1.85868	-0.04439	1.85921
TestingPoint_08	1.54137	0.16472	1.55014
TestingPoint_08	1.59730	-0.11761	1.60162
TestingPoint_08	1.86665	-0.05757	1.86753
TestingPoint_08	2.04523	-0.11946	2.04871

Table 14: The testing result at all test-points.

REFERENCES

[1] R. F. Brena, J. P. García-Vázquez, C. E. Galván-Tejada, D. Muñoz-Rodriguez, C. Vargas-Rosales, and J. J. J. o. S. Fangmeyer, "Evolution of indoor positioning technologies: A survey," vol. 2017, 2017.

[2] L. Batistić and M. Tomic, "Overview of indoor positioning system technologies," in *2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, 2018, pp. 0473-0478.

[3] A. Khalajmehrabadi, N. Gatsis, and D. Akopian, "Modern WLAN Fingerprinting Indoor Positioning Methods and Deployment Challenges," *IEEE Communications Surveys & Tutorials,* vol. 19, no. 3, pp. 1974-2002, 2017.

[4] K. I. Gettapola, R. R. W. M. H. D. Ranaweera, G. M. R. I. Godaliyadda, and M. N. F. Imara, "Location based fingerprinting techniques for indoor positioning," in *2017 6th National Conference on Technology and Management (NCTM)*, 2017, pp. 175-180.

[5] B. Alavi and K. Pahlavan, "Modeling of the distance error for indoor geolocation," in *2003 IEEE Wireless Communications and Networking, 2003. WCNC 2003.*, 2003, vol. 1, pp. 668-672 vol.1.

[6] J. Mier, A. Jaramillo-Alcázar, and J. J. Freire, "At a Glance: Indoor Positioning Systems Technologies and Their Applications Areas," Cham, 2019, pp. 483-493: Springer International Publishing.

[7] P. Le Van Hoang, V. T. J. S. Quang, T. D. Journal-Engineering, and Technology, "BLE-based Indoor Positioning System for Hospitals using MiRingLA Algorithm," vol. 2, no. 1, pp. 46-59, 2019.

[8] G. Deak, K. Curran, and J. Condell, "A survey of active and passive indoor localisation systems," *Computer Communications,* vol. 35, no. 16, pp. 1939-1954, 2012/09/15/ 2012.

[9] Q. Liu, J. Qiu, and Y. Chen, "Research and development of indoor positioning," *China Communications,* vol. 13, no. 2, pp. 67-79, 2016.

[10] M. S. Chebli, H. Mohammad, and K. A. Amer, "An Overview of Wireless Indoor Positioning Systems: Techniques, Security, and Countermeasures," Cham, 2019, pp. 223-233: Springer International Publishing.

[11] "Wi-Fi Generations," *WiFi Alliance*, pp. https://www.wi-fi.org/discover-wi-fi, 2020.

[12] "Radio Versions," *Bluetooth*, pp. https://www.bluetooth.com/learn-about-bluetooth/bluetooth-technology/radio-versions/, 2020.

[13] P. Martins, M. Abbasi, F. Sá, J. Cecílio, F. Morgado, and F. Caldeira, "Improving bluetooth beacon-based indoor location and fingerprinting," *Journal of Ambient Intelligence and Humanized Computing*, 2019.

[14] F. Zafari, A. Gkelias, and K. K. Leung, "A Survey of Indoor Localization Systems and Technologies," *IEEE Communications Surveys & Tutorials,* vol. 21, no. 3, pp. 2568-2599, 2019.

[15] F. Dwiyasa and M. Lim, "A survey of problems and approaches in wireless-based indoor positioning," in *2016 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, 2016, pp. 1-7.

[16] S. Xia, Y. Liu, G. Yuan, M. Zhu, and Z. J. I. I. J. o. G.-I. Wang, "Indoor fingerprint positioning based on Wi-Fi: An overview," vol. 6, no. 5, p. 135, 2017.

[17] K. Nguyen and Z. Luo, "Conformal Prediction for Indoor Localisation with Fingerprinting Method," Berlin, Heidelberg, 2012, pp. 214-223: Springer Berlin Heidelberg.

[18] K. A. Nguyen, "A performance guaranteed indoor positioning system using conformal prediction and the WiFi signal strength," *Journal of Information and Telecommunication,* vol. 1, no. 1, pp. 41-65, 2017/01/02 2017.

[19] N. u. H. Lodhi, A. Malik, T. Zulfiqar, M. A. Javed, and N. S. Nafi, "Performance Evaluation of Wi-Fi Finger Printing based Indoor Positioning System," in *2018 IEEE Conference on Wireless Sensors (ICWiSe)*, 2018, pp. 56-61.
[20] Y.-C. Pu and P.-C. J. A. M. M. You, "Indoor positioning system based on BLE location fingerprinting with classification approach," vol. 62, pp. 654-663, 2018.
[21] S. Subedi and J.-Y. J. J. o. S. Pyun, "Practical fingerprinting localization for indoor positioning system by using beacons," vol. 2017, 2017.
[22] E. Teoman and T. Ovatman, "Trilateration in Indoor Positioning with an Uncertain Reference Point," in *2019 IEEE 16th International Conference on Networking, Sensing and Control (ICNSC)*, 2019, pp. 397-402.
[23] M. E. Rusli, M. Ali, N. Jamil, and M. M. Din, "An Improved Indoor Positioning Algorithm Based on RSSI-Trilateration Technique for Internet of Things (IOT)," in *2016 International Conference on Computer and Communication Engineering (ICCCE)*, 2016, pp. 72-77.
[24] Y. Zhang, *New Advances in Machine Learning*. BoD–Books on Demand, 2010.
[25] S. B. Kotsiantis, I. Zaharakis, and P. J. E. a. i. a. i. c. e. Pintelas, "Supervised machine learning: A review of classification techniques," vol. 160, pp. 3-24, 2007.
[26] T. Arvind Kumar, "Introduction to Machine Learning," in *Ubiquitous Machine Learning and Its Applications*, K. Pradeep and T. Arvind, Eds. Hershey, PA, USA: IGI Global, 2017, pp. 1-14.
[27] T. O. J. N. a. i. m. l. Ayodele, "Types of machine learning algorithms," pp. 19-48, 2010.
[28] D. Nastac, F. A. Iftimie, O. Arsene, V. Ilian, and B. Cramariuc, "Indoor positioning WLAN based fingerprinting as supervised machine learning problem," in *2017 IEEE 23rd International Symposium for Design and Technology in Electronic Packaging (SIITME)*, 2017, pp. 194-199.
[29] MathWorks, "What Is Machine Learning?," pp. https://au.mathworks.com/discovery/machine-learning.html, 2020.
[30] S. Yiu, M. Dashti, H. Claussen, and F. Perez-Cruz, "Wireless RSSI fingerprinting localization," *Signal Processing*, vol. 131, pp. 235-244, 2017/02/01/ 2017.
[31] R. Bruha and P. Kriz, "Different Approaches to Indoor Localization Based on Bluetooth Low Energy Beacons and Wi-Fi," Cham, 2017, pp. 305-314: Springer International Publishing.
[32] M. Petric, A. Neskovic, and N. Neskovic, "Dynamic k nearest neighbours model for mobile user indoor positioning," in *2015 23rd Telecommunications Forum Telfor (TELFOR)*, 2015, pp. 165-168.
[33] D. Herrmann, R. Wendolsky, and H. Federrath, "Website fingerprinting: attacking popular privacy enhancing technologies with the multinomial naïve-bayes classifier," in *Proceedings of the 2009 ACM workshop on Cloud computing security*, 2009, pp. 31-42.
[34] B. Li, A. G. Dempster, J. Barnes, C. Rizos, and D. Li, "Probabilistic algorithm to support the fingerprinting method for CDMA location," in *Int. Symp. On GPS/GNSS*, 2005, pp. 8-10.
[35] K. Mekki, E. Bajic, F. Chaxel, and F. Meyer, Concept and Hardware Considerations for Product-Service System Achievement in Internet of Things. 2019.
[36] JIMBLOM, "ESP32 Thing Hookup Guide," pp. https://learn.sparkfun.com/tutorials/esp32-thing-hookup-guide#introduction, 2019.
[37] Arduino, "Software," p. https://www.arduino.cc/en/main/software, 2020.
[38] Arduino, "WiFi library," p. https://www.arduino.cc/en/Reference/WiFi, 2019.
[39] PuTTY, "PuTTY: a free SSH and Telnet client," p. https://www.chiark.greenend.org.uk/~sgtatham/putty/, 2019.
[40] MathWorks, "What is MATLAB?," pp. https://au.mathworks.com/discovery/what-is-matlab.html, 2020.
[41] MathWorks, "Statistics and Machine Learning Toolbox," p. https://au.mathworks.com/help/stats/, 2020.